DOCUMENT RESUME

ED 142 180 IR 004 971

AUTHOR Marcus, Richard S.; Reintjes, J. Francis
TITLE Computer Interfaces for User Access to Heterogeneous Information-Retrieval Systems.
INSTITUTION Massachusetts Inst. of Tech., Cambridge. Electronic Systems Lab.
SPONS AGENCY National Science Foundation, Washington, D.C.
REPORT NO ESI-R-739
PUB DATE Apr 77
GRANT NSF-SIS-75-22946
NOTE 81p.; Pages 63-72 illegible for reproduction.

EDRS PRICE MF-$0.83 HC-$4.67 Plus Postage.
DESCRIPTORS *Bibliographic Citations; Computer Assisted Instruction; Computer Oriented Programs; *Information Retrieval; Information Systems; *Input Output Analysis; *Man Machine Systems; On Line Systems; Use Studies

ABSTRACT

A translating-computer-interface approach to providing a common, or virtual-system, mode of access to a network of heterogeneous online bibliographic retrieval systems has been investigated. Enhanced access to such systems by end users has been demonstrated through test usage of an experimental interface. A table-driven, rule-based message interpreter that provides a dynamic and flexible means for handling the automation of the interconnection of computer and human retrieval protocols has been found useful in implementing the interface. Experimental analysis has demonstrated the success of various computer-assisted instructional techniques as well as indicated basic tradeoffs and limitations in their implementation in working systems. Further development of interface systems appears promising and possibilities for further research have been outlined in a number of areas. (Author)

****
* Documents acquired by ERIC include many informal unpublished *
* materials not available from other sources. ERIC makes every effort *
* to obtain the best copy available. Nevertheless, items of marginal *
* reproducibility are often encountered and this affects the quality *
* of the microfiche and hardcopy reproductions ERIC makes available *
* via the ERIC Document Reproduction Service (EDRS). EDRS is not *
* responsible for the quality of the original document. Reproductions *
* supplied by EDRS are the best that can be made from the original. *
****

ED 142 188

April, 1977

U S DEPARTMENT OF HEALTH, EDUCATION & WELFARE
NATIONAL INSTITUTE OF EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

Report ESL-R-739

COMPUTER INTERFACES FOR USER ACCESS TO HETEROGENEOUS INFORMATION - RETRIEVAL SYSTEMS

by

Richard S. Marcus
J. Francis Reintjes

This material is based upon research supported by the National Science Foundation under Grant No. NSF-GIS-75-22946. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Electronic Systems Laboratory
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

TR004911

ABSTRACT

A translating-computer-interface approach to providing a common, or virtual-system, mode of access to a network of heterogeneous online bibliographic retrieval systems has been investigated. Enhanced access to such systems by end users has been demonstrated through test usage of an experimental interface. A table-driven, rule-based message interpreter that provides a dynamic and flexible means for handling the automation of the interconnection of computer and human retrieval protocols has been found useful in implementing the interface. Experimental analysis has demonstrated the success of various computer-assisted instructional techniques as well as indicated basic tradeoffs and limitations in their implementation in working systems. Further development of interface systems appears promising and possibilities for further research have been outlined in a number of areas.

ACKNOWLEDGEMENTS

This report constitutes the final report for Grant NSF-SIS-75-22946 entitled "Research in the Coupling of Interactive Information Systems." The project was supported by the Division of Science Information of the National Science Foundation. The grant period ran from November 1, 1975 to April 30, 1977.

We wish to acknowledge the cooperation extended to us by the National Library of Medicine, the Lockheed Corporation, and the System Development Corporation in conjunction with our use of their respective online retrieval systems. Recognition of contributions made by our staff and students who assisted us in the project is also made.

CONTENTS

		Page
1.	INTRODUCTION	1
	1.1 The Need for Improved Access to Retrieval Systems	1
	1.2 The Translating-Interface/Virtual-System Approach	3
	1.3 Outline of Current Work	4
2.	ADVANCED CONIT EXPERIMENTAL SYSTEM	7
	2.1 Physical Interconnections to Remote Systems	7
	2.1.1 Initial Techniques	7
	2.1.2 The Auto-Call Connection	9
	2.1.3 Current and Future Status of Auto-Call	10
	2.2 A Rule-Based Message Interpreter	10
	2.2.1 Background	10
	2.2.2 The Rule-Table Approach	11
	2.2.3 Rule Ordering: Longest-Match Principle	13
	2.2.4 Rule Structures and Special Action Types	15
	2.2.5 Rule Contexts: Generality Principle	17
	2.2.6 Explanations of Example Rules	18
	2.2.7 Significance of the Rule Interpreter	25
	2.3 User Functions in CONIT	26
	2.3.1 Overview	26
	2.3.2 System and Data-Base Selection and Connection	27
	2.3.3 Additional User-Selected Instruction	28
	2.3.4 Search and Index Operations	28
	2.3.5 Document Reference Output	29
	2.3.6 Interrupting and Editing	29
3.	EXPERIMENTATION	30
	3.1 Overview of the Experimental Program	30
	3.2 An Example of a User Session	31
	3.3 Interpretations of Test Usages	37

Page

3.3.1 Overall Observations 37
3.3.2 General User Psychology and Interface Design 38
3.3.3 Data-Base Explanation and Selection 42
3.3.4 Other Interface Considerations 43

4. SUMMARY 45

5. THE FUTURE OF THE VIRTUAL-SYSTEM CONCEPT 46

6. PROJECT BIBLIOGRAPHY 47

7. REFERENCES 48

APPENDIX A. Explanation of Context Strings 50
B. Listing of Special Actions 52
C. EXPLAIN Command Messages 54
D. Interactive Translation Rules 62
E. Excerpts from Experimental Session Dialogs 74

1. INTRODUCTION

1.1 The Need for Improved Access to Retrieval Systems

As Burchinal[1*] has pointed out, the use of interactive bibliographic retrieval systems has increased significantly in recent years. This application is thus starting to fulfill its early promise as one of the important applications to be served by the growing field of computer-based time-shared systems. More than a million searches are performed annually by a number of different systems which have access, in the aggregate, to several score data bases containing, in total, more than 20 million references to documents of many different types -- e.g., books, reports, journal and news articles, etc. -- in a wide range of subject areas in science, technology, business, law, and the arts. There has been a steady rise in these statistics over the last few years as new systems and data bases have come online and more and more users have learned of their existence and retrieval effectiveness.

The very success of these systems has tended to aggravate the problem of convenient usage because of the difficulties encountered by users in learning how to interact with the multiplicity of heterogeneous systems and data bases. A potential user of different retrieval systems is faced with a series of obstacles right from the start: the necessity to discover these systems in the first place, to enter into separate procedures to gain access and reimburse costs, and possibly -- if the systems are not interconnected through a common network -- to make actual access via different terminals and separate locations. Other obstacles face the user once the initial access is made: different command languages, retrieval functions, indexing vocabularies, and output formats. Even within a given system, access to different data bases is often frustrated by the differences in catalog record fields and indexing methods that the system may only partially compensate for. It is little

---

* Superscripted numerals refer to references listed in Section 7.

wonder, then, that currently access to these systems is primarily through a professional intermediary -- a specially trained librarian, for example, rather than by the user himself.

It might be thought that a single system and data base should satisfy a given user. It has been our experience at M.I.T. with the M.I.T. Intrex[2], Lockheed DIALOG[3], Systems Development Corporation (SDC) ORBIT[4], ERDA RECON[5], The New York Times Information Bank[6], and MEDLINE[7] systems, however, that a single user generally needs access to many different bases, if not for a given search, then over a period of time as his needs change. Furthermore, in a community of professionals with heterogeneous interests, it may be the case that the collection, storage, and access to a multiplicity of resources are better arranged through separate data bases rather than through aggregation into a single huge data base.

The differences in the retrieval systems present substantial difficulties even to experienced users. In the NASIC at M.I.T. program[8], where librarians have been trained as information specialists to assist end users in searching online data bases, we have found that several weeks of training and continuing practice at the terminal were needed by the specialists to get to a high level of proficiency and to maintain that level. A significant part of the learning difficulty was caused by the differences among data bases and systems. Even the specialists have found it desirable to specialize in a small number of data bases and, sometimes, in only one or two systems, at least partly for the reason of the heterogeneity of data bases and systems. Another reason, of course, is that existing systems have not yet realized the full potential of computer-assisted instruction.

The end user* may not have to master as many data bases and systems as his specialist intermediary, but this contraction is more than offset by the fact that, in general, the end user has neither the time nor

---

* By end user we mean the person who has a direct need for the information being sought. An end user is a seeker of information who is frequently unskilled in the use of online retrieval systems.

the inclination for much training or practice. In fact, it is for this reason that NASIC and others have decided that it is unrealistic in the present information-retrieval environment to expect end users to do their own searches, especially when the computer time -- as well as the user's time -- is a costly commodity.

1.2 The Translating-Interface/Virtual-System Approach

In order to investigate means to surmount obstacles hindering convenient and effective use of the multiplicity of heterogeneous interactive bibliographic retrieval systems, the M.I.T. Electronic Systems Laboratory has undertaken a research program to examine the feasibility of interconnecting interactive retrieval systems through computer interfaces. The computer interface would achieve compatibility among systems of heterogeneous hardware and software components through use of common retrieval protocols or by translating dissimilar protocols to a common set. (See Fig. 1.)

Our research program emphasizes an approach in which the interface is, in effect, a common system into which, and from which, requests and results are translated automatically as they flow between user and serving systems. This approach has the virtue that a user attempting to retrieve information, when entering through the access mechanism provided by the common interface, sees a single virtual system in which all the complexities of the different retrieval systems and data bases are hidden; only a single uniform system is apparent. In this way the goal of convenient use of heterogeneous computer resources is achieved, at least for the particular application of interactive bibliographic retrieval systems. There are three aspects of our approach which, taken together, distinguish our efforts to achieve networking: (1) We are concentrating on the information transfer application, with particular emphasis on online bibliographic retrieval systems; (2) we utilize existing, major, stand-alone interactive systems without modification; and (3) we are placing emphasis on serving the ordinary end user -- that is, a user experienced neither in computer programming, general

computer usage, nor in the use of interactive retrieval systems.

Our research to date on this project has included investigations of the following related areas:

(1) The parameters involved in evaluating interactive interfaces

(2) Human factors and instructional considerations

(3) Common retrieval languages

(4) Indexing conversion techniques

(5) Commonality among data-base structures

(6) Logical structures for networked interfaces

(7) Experimental interface systems

(8) User evaluation with an experimental interface

(9) Cost/benefits tradeoffs

Our work in these areas has been detailed in previous reports and papers (See Section 6: Project Bibliography).

1.3 Outline of Current Work

In the rest of this report we describe the work that has been performed during the period of this grant, November, 1975 through April, 1977. In Section 2 we describe the development of an advanced experimental interface for investigating various aspects of the virtual-system/translating-computer-interface concept. This development has encompassed three aspects:

(1) The implementation of an auto-call mechanism for connecting from the interface to remote systems.

(2) The development and implementation of a dynamic, table-driven, rule-based, context-sensitive message interpreter and translator.

(3) The development of an advanced experimental interface based on the auto-call and translation-table mechanisms. The interface is designated CONIT, which stands for Connector for Networked Information Transfer.

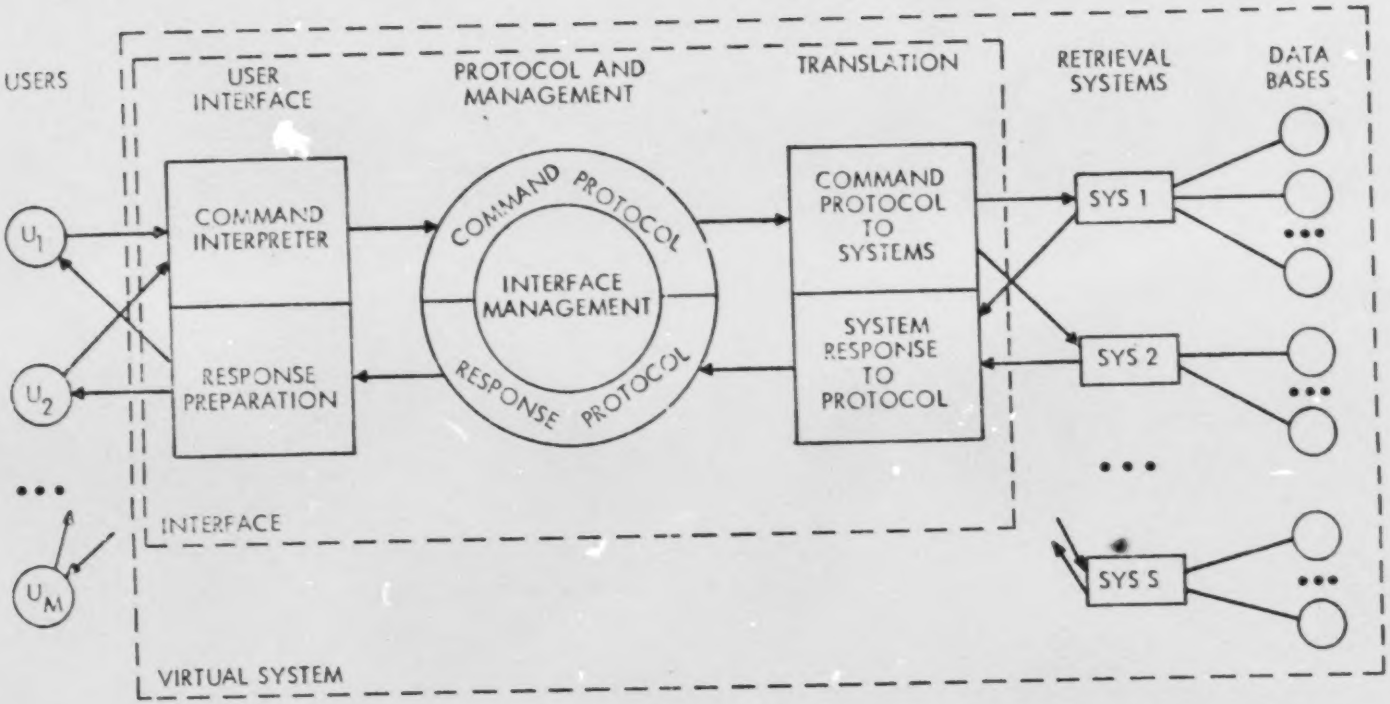

Fig. 1 Logical Diagram of Virtual-System Interface

In Section 3 we describe the testing and analysis of our latest interface techniques and concepts. This includes test usage by end users and analysis of how well our techniques satisfy the goals of easy and effective access. Sections 4 and 5 contain a summary of our work to date and an appraisal based upon this work of means by which the virtual-system concept may be further developed.

Sections 6 and 7 contain a project bibliography and references for this report, respectively. The appendices contain computer printouts, or material derived therefrom, with detailed information on CCNIT trials, instructional messages, translation-rule tables, and explanations of rule components.

2. ADVANCED CONIT EXPERIMENTAL SYSTEM

In our previous work[P6*] we had demonstrated -- partly through experimentation with an initial version of CONIT -- the feasibility of the interface concept for interconnecting interactive bibliographic retrieval systems. The main emphasis of our research during the current grant period has been to investigate further whether this concept has practical utility, especially an ability to provide naive end users direct, convenient access to information from the heterogeneous systems and data bases.

An important vehicle for this research has been the CONIT experimental interface which has been significantly expanded in order to make it more useful as a research tool.

2.1 Physical Interconnections to Remote Systems

2.1.1 Initial Techniques

Our initial mode of interconnection to remote systems involved use of the ARPANET -- a packet-switched computer network developed under sponsorship of the Advanced Research Projects Agency (ARPA) of the U.S. Department of Defense. In particular, use was made of the ARPANET TIP -- the Terminal IMP (Interface Message Processor) minicomputer which provides the basic communications subnet of ARPANET. TIPS interconnect the diverse host computers and also give end users access to the host computers through a dial-in procedure to standard telephone lines from any of a variety of computer terminals. By appropriately adjusting TIP terminal-port parameters, one can[P1, P4] also connect a remote host computer to a TIP terminal port, even though the host computer is not otherwise connected to the ARPANET. Through such connections we initially communicated between our first experimental interface, which was located on an ARPANET host computer (the M.I.T. MULTICS System), and several retrieval systems (See Fig. 2).

---

* "P-number" references refer to documents in the Project Bibliography (See Section 6).

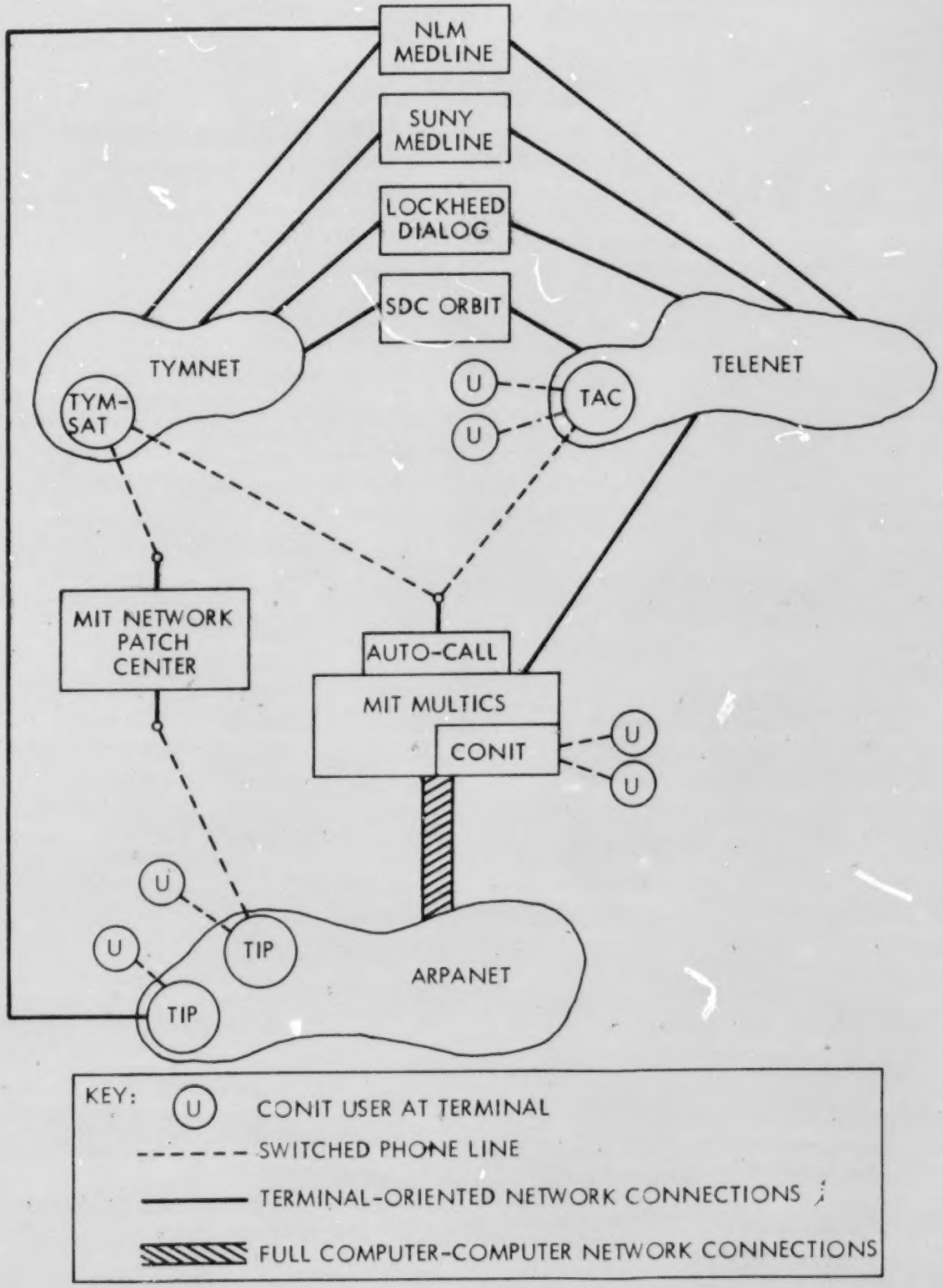

Fig. 2 Computer and Network Interconnections for CONIT Experimental Interfaces

Although only low-speed communication was provided by these TIP terminal-port connections, we were able to try out several preliminary virtual-system/translating-interface techniques and prove feasibility of the logical concept, at least at a basic level. There are, however, two deficiencies in this kind of connection. First, it involves using the TIP for a purpose for which it was not intended; this unorthodox, nonstandard mode of connection suffers from a difficulty and unreliability of operation, caused, in part, by unanticipated changes in TIP software or hardware. Second, these connections were either limited to hard-wired leased lines or required an awkward manual set-up operation to "patch" two dialed lines together.

2.1.2 The Auto-Call Connection

To permit meaningful experimental investigations, we needed a better form of data communications. This was achieved during this grant period in the form of an "Auto-Call" device attached to the MULTICS system. The Auto-Call mechanism allows MULTICS, under computer program control, to call up another computer over regular switched-telephone-line circuits.

The Program Development Office of the M.I.T. Information Processing Center (IPC) performed the modifications to the M.I.T. Honeywell 6180-based MULTICS system that were required to incorporate the Auto-Call facility. An outline of these modifications, as excerpted from IPC's documentation[9-10], follows:

1. Hardware Requirements
   - a. Bell Auto-Call Unit 801A ACU
   - b. Bell 103A Modem
   - c. Honeywell HSC351 Logic Board for the High-Speed Line Adapter (HSLA) of I/O Controller (DN355)

2. Software Modifications
   - a. Honeywell 355 code
   - b. MULTICS Hard-Core and User-Ring TTY Software
     Two new "order" operations

i. Initiate auto-call

ii. Obtain status of auto-call channel

c. Initializer (Answering Service) Software

3. Reconfiguration of System Status Tables

a. Configuration Cards

b. Channel Master File and Definition Table

c. Channel Access Control

All control operations (line assignment, telephone dialing, and hangup processing) are performed by the Initializer in response to user-process-generated Inter-Process Communication requests or to hardware conditions. After a line has been assigned, the user process may perform input and output over the line using standard MULTICS I/O commands.

2.1.3 Current and Future Status of Auto-Call

The installation of the Auto-Call facility has permitted us to extend our experimental investigations as described in the remainder of the report. However, Auto-Call type communications should be considered only as a temporary expedient until more standard high-speed network communications are available. This kind of improved communications will be achieved, for example, between our MULTICS interface and such TELENET network host retrieval systems as MEDLINE, SDC, Lockheed, the Bibliographic Retrieval Service (BRS), and the N.Y. Times when the MULTICS connection to TELENET is upgraded to provide computer-initiated outbound traffic in addition to the terminal-initiated inbound traffic. Nevertheless, Auto-Call type communications will continue to be useful in the forseeable future so long as some systems with which one might want to interconnect are outside a single network -- or group of several networks already interconnected by suitable hardware and software mechanisms.

2.2 A Rule-Based Message Interpreter

2.2.1 Background

In our previous report[P6] we described the development of a model by

which one can characterize the interaction of independent and heterogeneous processes involved with interface operations. One special character of this interaction derives from the fact that although the messages coming from the retrieval systems were designed for human interpretation, in this situation they are actually interpreted by a computer process: the interface. We felt there was a need, then, that the model should point to the design of a software structure that would provide an effective, flexible mechanism for implementing a major part of the message interpretation, translation, and response functions inherent in this kind of computer-mediated, but human-oriented, message interchange. The outline of such a structure was developed and is reproduced here as Fig. 3.

Our initial efforts[P3, P6] at software structures for message handling and translation left a lot to be desired as far as effectiveness and flexibility are concerned. Different functions -- e.g., host login, host logout, user command translation -- were performed by entirely different program modules using different structures. Only the command and response translation functions were table-driven and these tables performed only very simple string-string translations. There were separate tables for each host system and for user command translation and system response translation; this modularity was desirable but achieved at the expense of considerable redundancy in that translation rules that were identical for different systems had to be repeated in the separate tables for those systems.

2.2.2 The Rule-Table Approach

During this reporting period we have designed and implemented a table-driven interpreter for handling the bulk of the message-interaction functions for our translating-interface situation. The structure of this interpreter closely follows the interface-model structure we had previously developed, as shown in Fig. 3. The general concept is that each action by the interface is governed by a table of rules. A rule is selected according to the current context (i.e., situation) and what is currently found in either (a) the user input stream (buffer), or (b) the response

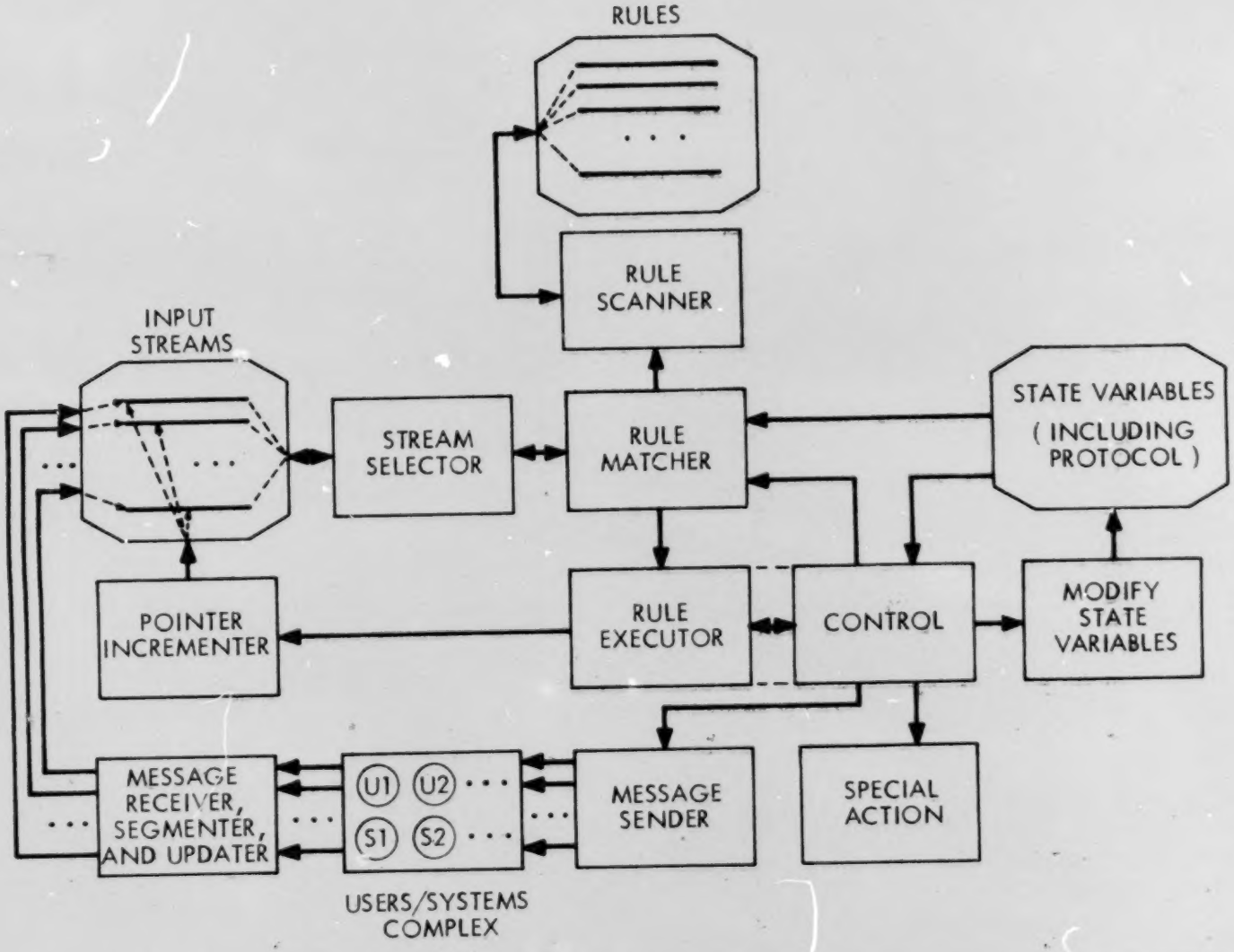

Fig. 3 Model of Message Interpretation and Response Components in Interface

input stream for the currently connected host retrieval system. The actions that result from the selection of a rule are of four types:

(1) Sending message(s) to user and/or connected host

(2) Resetting the context description

(3) Incrementing the pointer to the "current position" in the input buffer

(4) Performing some "special action" from a set of such actions pre-defined as subroutines

Incrementing the buffer pointer, action type 3, is accomplished according to how much of the character string in the buffer was specified in the rule selection. Any combination of the other 3 action types is performed in accordance with the specifications supplied in the selected rule. The general structure of the CONIT program is shown in flow-chart form in Fig. 4.

Specifically, each rule has a maximum of six parts:

1. context string (CS): a character string that specifies the context or situation
2. match string (M_b): a character string to be matched in the designated input stream
3. host message (HM): message to send to host system
4. user message (UM): message to send to user
5. new context (NC): revised context put in effect after rule execution
6. special action (SA): particular function to be performed

2.2.3 Rule Ordering: Longest Match Principle

The rules are arranged in a particular order corresponding to the order in which they are scanned to find the first rule that matches the current context (i.e., the new context of the last rule executed) and the designated input stream. For each different context string the rules are ordered in the inverse alphabetical order of their match strings. If two strings are different lengths, and the shorter one is a prefix of the

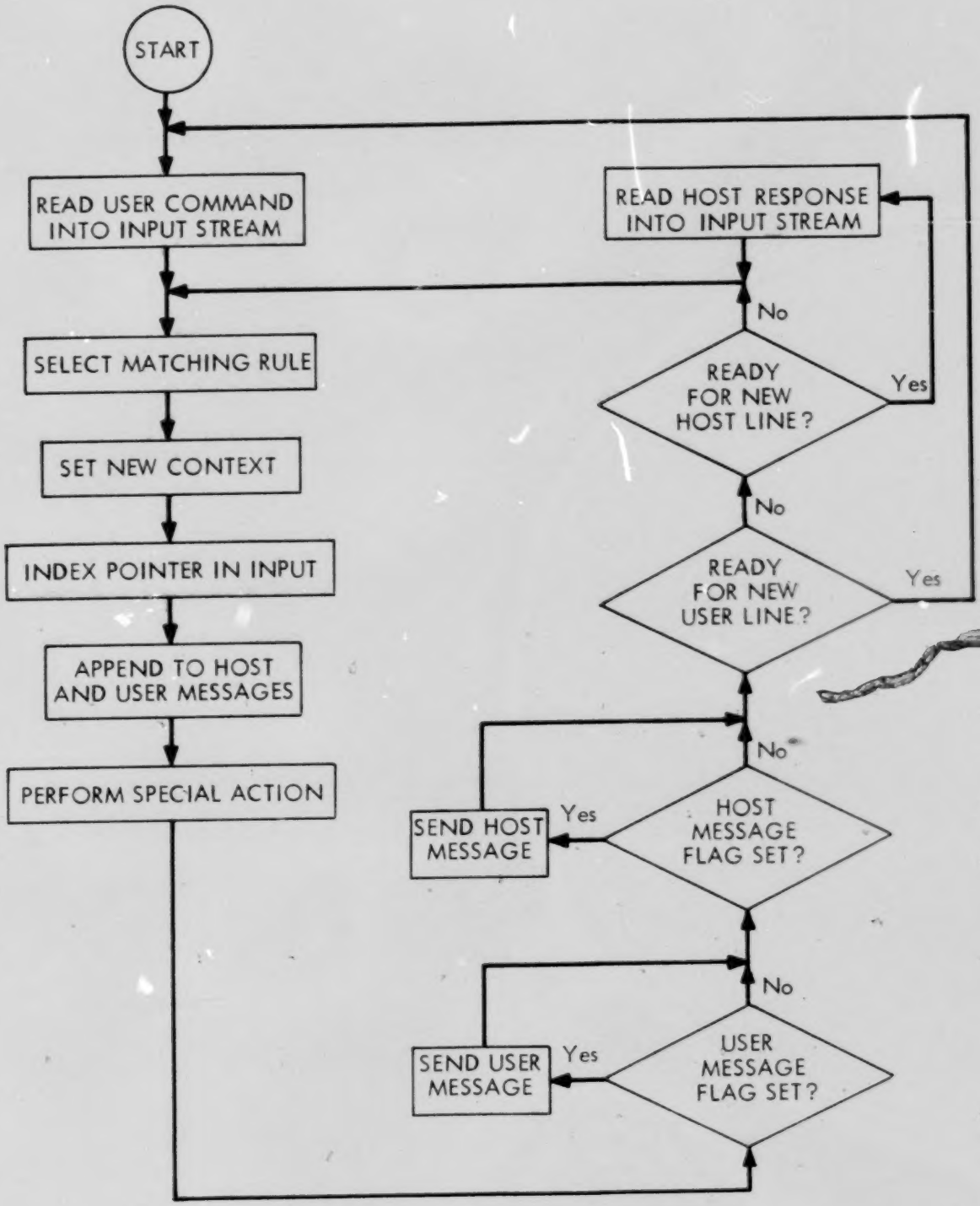

Fig. 4 Flow Diagram of CONIT Experimental Program

83492AW004

longer one, then the longer one is ordered first; this provides for the longest match principle: longer matches are selected before shorter ones. Finally, rule-ordering is made definitive by stating that context strings are also ordered in the inverse-alphabetical, longest-first order.

2.2.4 Rule Structures and Special Action Types

Rules are maintained in a list structure as schematized in Fig. 5. This structure allows easy addition to, or modification of, the rules or their component parts.

A particular subset of the special actions extend the logical power of the rule interpreter in a significant way. This subset performs operations on a set of numerical variables. These numerical variables are used to represent states of the interaction that are not conveniently representable in the context, which is a relatively short alphanumeric character string. The operations include: (1) setting a particular variable as specified by the number in the user message component of this rule or by a numerical character string in the input stream; or (2) inserting a numerical character string into either the host message, the user message, or the input stream. This last operation of inserting into the input stream extends the capabilities of the rule interpreter so as to include those normally associated with a Turing Machine, although, of course, only a finite-sized input stream is accessible to a realizable rule interpreter. A more complete description of all special actions is given in Appendix B.

Normally, only part of the input stream is matched by the match string and the next rule is sought after incrementing the pointer in the input buffer to the end of the match and appending the host and user message rule parts to the corresponding host and user output buffers. Normally, neither the host nor user messages are actually sent at this point. Another major subset of the special actions is needed to control the interpreter operations when the rule detects a situation where messages should be sent and/or new information is sought for the input buffer. Such a situation might arise, for example, when the input buffer

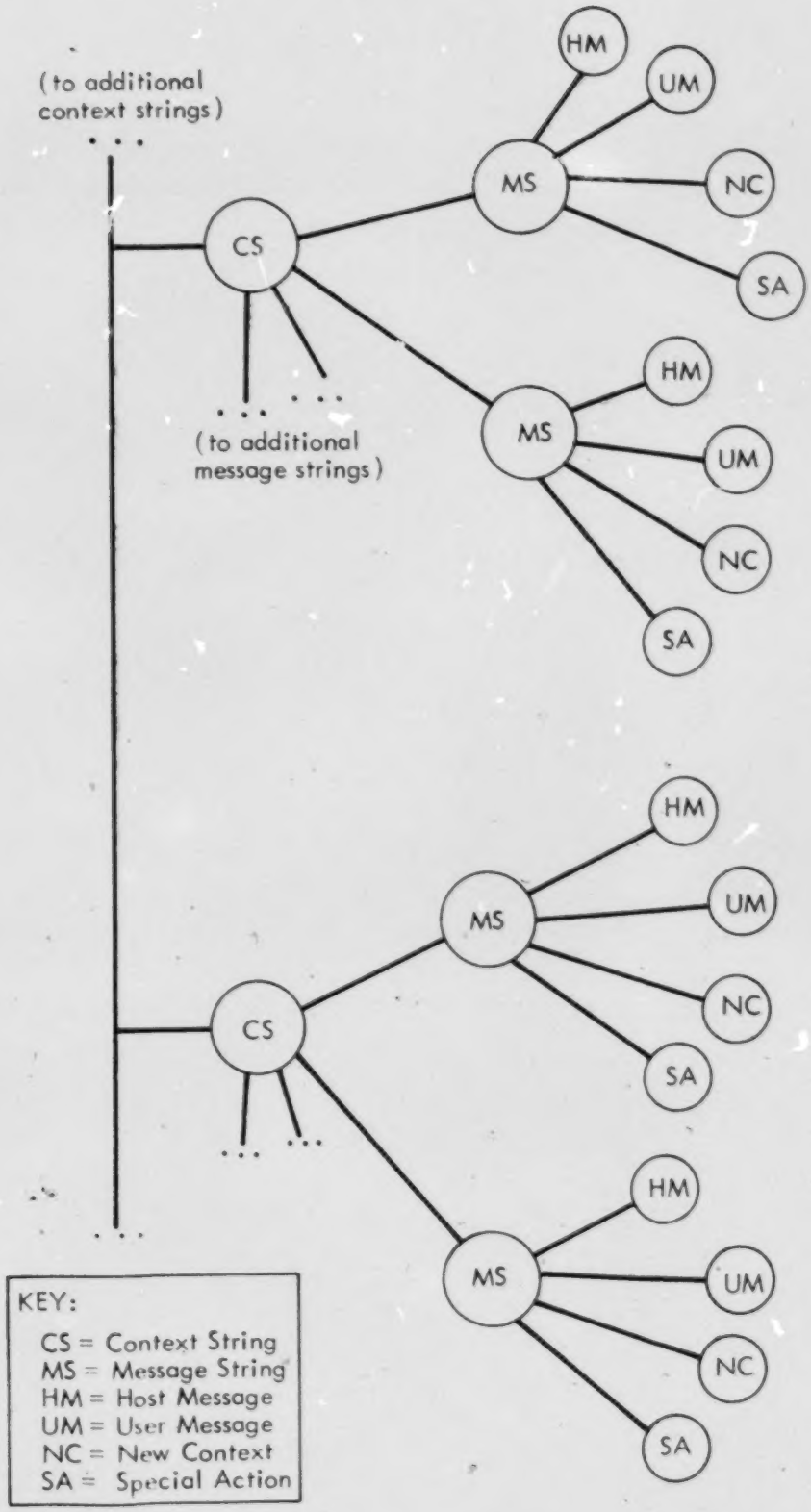

Fig. 5 Structure of CONIT Translation Rule File

83492AW005

is exhausted by the match string. The detailed description of these special actions is also given in Appendix B.

2.2.5 Rule Contexts: The Generality Principle

The context string is an alphanumeric character string having a minimum of 7 characters. Each position in the string is reserved for a particular contextual meaning. For example, the first character is reserved to indicate whether the current operation of the interpreter is based on the interpretation of user commands (character "u") or on the interpretation of a retrieval system response (character "r"). Other character positions are reserved to indicate such contextual modes as: the connection mode to a retrieval system (e.g., logging in, connected, logging out, or detached); the network being used (e.g., Telenet, Tymnet, or none); the system connected to (e.g., Lockheed, SDC, NLM MEDLINE, or SUNY MEDLINE); the particular user command being processed; the mode of processing for the current command (e.g., parsing user's command line, translating command to remote system language, or interpreting and translation system response); the sequence number with the series of protocol steps for the current operation; and the presence of a particular "special" situation (e.g., the communication line to the remote host has been hung up, more than a pre-specified amount of time has elapsed waiting for a host system response, a break signal from the user has been detected, etc.). A detailed description of the currently used context modes is given in Appendix A.

An important feature of the algorithm for selecting the appropriate rule in the current context is that the context-string component for one rule may match on many context strings. This many-to-one mapping is achieved in two ways. First, the rule context-string component may be truncated to fewer than 7 characters. Any context string that has a prefix matching the context-string component is considered to match. Second, any character positions in the rule-context component may assume a match-all character -- the hyphen (-) -- such that any characters in those positions for the current context are considered to match the rule

context. The effect of these two assumptions is that one rule can serve for many combinations of contextual situations where we want to consider all these combinations as equivalent. That, in turn, means that the number of rules need not be overly large, although the number of different contextual situations may be enormous because of the combinatorial factor.

It is important to note that the inverse-alphabetical, longest-first ordering principle causes the more generic rules -- i.e., those with truncated context components or with match-all hyphens -- to sort after the more specific rules (ASCII hyphen sorts after the alphabet in inverse order). Thus, the rules with more specific context components take precedence, where they exist and match; otherwise, the more general rules take over by default. A subset of the rules for the current experimental interface in the longest-match, more-specific first order is shown in Fig. 6.

2.2.6 Explanations of Example Rules

The full set of approximately 500 rules for the advanced CONIT interface is given in Appendix D. In this subsection we have extracted twenty-five rules from the complete set to illustrate the several features of the rules which we have described above. These 25 rules are representative of one cycle of events that occur in the processing of a single user command (the SHOW command) and the resultant system responses. The functional capabilities of the advanced CONIT which has been implemented in accordance with these rules -- that is, CONIT as seen by the user -- are presented in Section 2.3.

Rules are listed according to the following format:

/<CS>/<MS>:.::<HM>::,,<UM>,,//<NC>//..<SA>..**<CM>

where

CS = Context String
MS = Match String
HM = Host Message
UM = User Message
NC = New Context

```
...
/rlog/ NOT !.//rlq1xnn//
/rlog/ BUSY!.//rlq1xbs//
/riggy--7/!.//riogy!!7//
/rigge--7/!.//riogc!!l2//
/riel-inz-/!.//riog!onx!//
/riel---b/!.,,You are now connected to Lockheed DIALOG.,,//riog!!!h//..sendc..
/rdtt/!.//udd1+//
/rdto/!.//rdq1+//
/rdq1/!.,,logging off from the retrieval system has been aborted.,,//udo1!!!//..senduc
/rdety--7/0!.//riggyyy7//
/rdety--6/DROPPED BY HOST!.//rdet!!!7a//
/rdete--7/0!.//riggeee7//
/rdet-os4/GOOD-BYE!.,,Logged off SUNY/MEDLINE.,,//rdet!!!6//..send..
/rdet-oo5/!.!!!!,,Logged off SDC/ORBIT.,,//rdet!!!6//..rsendnnl..
/rdet-oo4/PLEASE HANG UP!...send..
/rdet-oo4/GOOD-BYE!.//rdet!!!5//
/rdet-oo4/..-user!!.//rdq1+//
/rdet-o-4/TIME!...send..
/rdet-o-3/..-USER!!.!!YES!!//rdet!!!4//..rsend..
/rdet-o-2/SEARCH?!.//rdet!!!3//..send..
/rdet-o-2/DONE?!.//rdet!!!3//..send..
/rdet-o-2/BEFORE YOU STOP!...send..
/rdet-o-1/!.!!"STOP!!//rdet!!!2//..rsend..
/rdet-nnz/..-MEDLINE IS AVAILABLE!.,,Logging of NLM/TSO.,,//rdet!nny//..send..
/rdet-nny/LOGGED OFF!.//riog!onym//
/rdet-nn1/!.!!!!//rdet!!!3//..rsend..
/rdet-l-3/LOGOFF!...send..
/rdet-l-3/..LOGOFF!...send..
/rdet-l-3/..-tc>!.//rdet!!!7a//
/rdet-l-3/..-LOGOFF!.//rdet!!!6//..send..
/rdet-l-3/..-!...copyr..
/rdet-l-2/User!.,,Logged of DIALOG. Costs for DIALOG use!,,//rdet!!!3//..send..
/rdet-l-2/          !.,,Logged off DIALOG. Costs for the use of last data base!,,//rde
/rdet-l-1/!.!!LOGOFF!!//rdet!!!2//..rsend..
/rdet---7a/!.,,1,,//rdet!!!7//..setnuml..
/rdet---7/7!.//udd1!!!//
/rdet---7/6!.//unotnnn//..exit..
/rdet---7/5!.//riggtnn7//
/rdet---7/4!.//rigg!!!7//
/rdet---7/3!.//riggtoo7//
/rdet---7/2!.//riggton7//
/rdet---7/1!.//riggtos7//
/rdet---7/0!.//udd1!!!//
/rdet---6/DISCONNECTED!.//rdet!!!7a//
/rdet---3/dropped by host!.//rdet!!!7a//
/rdet---3/..-tc>!.//rdet!!!7a//
/rdet---1a/!.//rdet!!!1//..tm4..
/rdet/dropped by host!.,,1,,//rdet!!!7//..setnuml..
/rdel/!.,,Coulon't log off DIALOG properly,,//udd1+//..senduc..
/rcto-zs/!.//uconios//..send..
/rcq4-ooz2/!.//rcq4!ooy//..send..
/rcq4-ooz/!.//rcq4!ooz2//..sendcr..
...
```

Fig. 6 An Ordered Subset of CONIT Translation Rules

SA = Special Action

CM = Explanatory Comment (not part of rule itself)

Let us now consider the following 25 example rules.

(1) /ucon/..-show:.//ucon:::shl//

Rule 1 is effective in the context where the user's command line is being interpreted and where a host retrieval system is already connected. The u that appears first in the context string signifies user command interpretation and the con indicates the existence of a connection to a host system. In this context we may have a rule match if the first four characters of the command line are "show"; the character string "..-" is appended to the beginning of each input line so we can distinguish the beginning of the line from other positions. When this rule is selected the sole immediate result is to establish a new context. The new context string begins with ucon. The next 3 characters of the new context string will be set the same as the corresponding 3 characters in the present context string -- signified by the 3 "retain" characters (:::) in the new context component; we thus retain the contextual modes for the current connected retrieval system and connecting network. Note that this rule may be selected whatever the current system and network since the context string component was truncated at 4 characters. The new context string is suffixed by the string shl which signifies that we are transferring to the first stage of that subset of rules that interprets (parses) parameters to the SHOW (citation outputting command). Note that for this rule no messages are sent or special functions performed.

(2) /ucon/..-show index :.//ucon:::sil//

Rule 2 is similar to Rule 1 but transfers context to the specialized SHOW INDEX command rather than to the more general SHOW (document citation) command. Similar rules exist for the SHOW NEWS and SHOW REVIEW commands. By longest-first ordering these rules come before Rule 1 and are selected first, when they apply, leaving SHOW (DOCUMENT) as the default case. This is convenient from a programming point of view since the 3 specialized

commands are easily distinguished by the first parameter, whereas SHOW (DOCUMENT) may have any one of about 6 parameters, including null, as its first parameter.

(3) /u/..- s :.,,..-show ,,..sub..

Rule 3 invokes a substitution into the input stream using the special action subroutine sub. It says: Whenever the command line begins with the character s followed by a space, replace that two-character string by its CONIT-synonymous five-character string show and reparse the line -- in which case Rule 1 or 2, etc., will apply. Since the context-string component is truncated to just u, this rule applies whatever the connection status. Note that for this special function the user-message rule component is used as an argument to the function which inhibits its role as a user message by appropriate flag settings (See SEND USER MESSAGE box in Fig. 4).

(4) /unot/..-show:.//unotrnc//

Rule 4 states that if the user issues the SHOW command when he is not connected to a system, signified by the substring not in the context, then control is transferred to the "error: not connected" context status (signified by substring rnc).

(5) /unotrnc/:.,,Conit cannot perform your request until you pick a system.,,//znotnnnpkex2//..senduc..

Rule 5 operates whenever the context is set to unotrnc -- as by Rule 4 -- whatever may be in the input stream. Note the null match string component. Special action senduc causes the given user message to be sent to the user and the interpreter control box (see Fig. 4) to go immediately to find the next matching rule (based on the new context). The new context defines the status:

a) error situation: z

b) no connection: notnnn

c) error message wanted is the second stage of the general message to explain command operation: ex2

d) for the PICK command: pk

(6) /ucon---sh1/:.::0::,,4,,//ucon:::sh2//..putnum..

Rule 6 states that the first stage of the parsing of the SHOW (DOCUMENT) command is to put the number (subroutine putnum) 0 (zero) in variable 4 and go on to stage 2. Variable 4 will be used to store the set number that the user specifies for showing; Rule 4, then, is a reset operation. Note that here again, message components are used for special-action argument purposes, not for messages. Note also that Rule 6 is effective whatever the current system connection (signified by "---").

(7) /ucon---sh5/set:.,,4,,//ucon:::sh5s//..getnum..

Rule 7 is selected at the fifth stage of show-command parsing if the string set is found next in the input stream (command line). In this case the routine getnum gets the numerical string expected next in the command line, converts it to a number, and stores it in variable 4.

(8) /ucnr---sh/:.//zcon:::shnr1//

Rule 8 is invoked when the getnum function fails to find the expected numerical string and inserts the substring nr (for numeral error) into the context. Context is transferred to the error procedure that tells the user his mistake (nr1). The substring sh in the new context retains the information that it was the SHOW command in which the mistake occurred.

(9) /ucon---sh5/s:.,,set,,..sub..

Rule 9 permits "s" to be a synonym for "set" as parameter to the SHOW command. Note that "s" can stand for either "show" or "set", depending on context.

(10) /ucon---sh5/ :.

Rule 10 allows an arbitrary number of spaces between parameters of the SHOW command. By this rule spaces are simply passed over.

(11) /ucon---sh5/-..:.//ucon:::sht1

Rule 11 detects the end-of-command delimiter string ("-..") and transfers control to the first stage of those rules that translate the now-parsed SHOW command to retrieval-system language.

(12) /ucon---sh5/:.//zcon:::shprl//

Rule 12 represents the default case at stage 5 of the SHOW command parsing routine: whatever is in the input stream matches nothing in the rules. For this error situation control is transferred to the set of rules that handles the explanation of parameter errors; this transfer is indicated by z in the first context position in conjunction with prl in the last 3 positions.

(13) /ucon---sht1/:.,,4,,//ucon:::sht5//..setnumi..

Stage 1 of the SHOW command translation calls on routine **setnumi** to set the number in variable 4 (as a character string) into the input stream and transfer control to stage 5.

(14) /ucon-o-sht5/:.::"PRT ::///ucon:o:sht6//

Rule 14 states that for the ORBIT family of systems (signified by o- and matched by oo: SDC/ORBIT, on: NLM/MDELINE, and os: SUNY/MEDLINE) the first part of the translation is the string '"PRT '.

(15) /ucon-l-sht5/:.::t::///ucon:l:sht6//

The corresponding rule for the Lockheed DIALOG system is to start the translation with the single character t. Related systems like NASA/RECON and ERDA/RECON would have similar translation rules.

(16) /ucon-o-sht6/0:.//ucon:o:sht2//

A 0 (zero) in the input means that variable 4 -- inserted by Rule 13 -- was not set in parsing. That, in turn, indicates that the user did not specify a set number. In this case, control simply passes to stage 2 of the translation.

(17) /ucon-o-sht6/:.::ss::,,4,,//ucon:o:sht2//..setnumo..

If the input contains anything other than zero, then the translation for the ORBIT retrieval language is to insert the string ss followed by the set number as given in variable 4. The setnumo routine performs the latter part of the operation.

(18) /ucon-l-sht6/0:.,,2,,//ucon:l:sht2//..setnumo..

(19) /ucon-l-sht6/:.,,4,,//ucon:l:sht2//..setnumo..

The corresponding rules for DIALOG cause only the user-given set number to be inserted (Rule 19) if there was one, or the number of the current retrieval set (from variable 2), otherwise (Rule 18).

(20) /ucon-o-sht9/:.//rcon:o:shr//..orbdoc..

In the ninth, and final, stage of SHOW command translation for the ORBIT systems, the routine orbdoc is called which inserts the string "skip n" into the translated command, where n=d1-1, and d1 is the number of the first document that the user has asked to see. "orbdoc" also calls on the rsend routine which sends the now fully-translated command to the host retrieval system. The initial r in the new context signifies that we are now going into "response" mode and that once the response is received, it will go into the input stream buffer where it will be analyzed in the show-response routine (shr in new context).

(21) /rcon---shr1..-:.:.copyr..

(22) /rcon---shr/..-TI -:.,,TITLE:,,//rcon:::shrh//

(23) /rcon---shr/..-USER::.//ucon:::shr2//

(24) /rcel-1-shr/:.//ucon:::shr2//

The basic rule in analyzing responses to the SHOW command is simply to copy each line (copyr routine) for the user (Rule 21). In the special case that the line begins with "-TI -" however, CONIT translates this to "TITLE:" before going on to stage h which copies the rest of the response line (Rule 22). The special line "USER:" is the cue (for ORBIT systems) that the SHOW response is completed, in which case control is sent to stage 2 in which instruction is given to the user on what can be done next (Rule 23). The cue that DIALOG finished its response is a period of 5 seconds without additional response; this is signalled through action of the response read routine inserting the substring el into the context (Rule 24). The read routine cannot "see" the actual DIALOG prompt "?" because it does not end with a new-line character and the MULTICS terminal I/O processor cannot read partial lines.

(25) /ucon---shr3/:.,,For explanation...,,//ucon:::///..send..

Rule 25 finishes off the explanation for the user and sends control to the point where the CONIT user prompt is given, and a new user command line is read and placed into the input stream for parsing.

2.2.7 Significance of the Rule Interpreter

Through the above examples and the details in the appendices, one can see that new message interpreter, which is the heart of CONIT, provides a high-level, application-oriented language in which to express the procedures for interfacing a user to heterogeneous, interactive information systems. This table-driven interpreter makes for easy and flexible preparation, debugging, documenting, reading, and modification of the rules of interaction. Since the rules can be modified online with CONIT itself, without recompilation, the interpreter is also highly dynamic and economical for making changes.

The new interpreter has permitted a streamlining and compaction of the previous CONIT programs. The current CONIT system, including translation-rule tables, occupies approximately 40 MULTICS storage pages (records) containing 1024 36-bit words each -- about half the size of the previous, less effective program. This size is compatible with minicomputer implementation, if that option should be considered.

The new message interpreter is clearly related in its structure to a number of recent so-called production systems, which are based on production rules.[11,12] Our interpreter appears different in detail from most of the other production systems in that it is specialized for the handling of message strings which are principal components of the rule selection process (also called input, context, left-hand side, data base, etc.) as well as of the rule action process (also called output, right-hand side, function, etc.)

Actually, the structure of this interpreter is more directly an outgrowth of work this author had done earlier[13,14] on the King originated work[15] on table look-up procedures for natural language processing which

has been the basis for machine-translation as well as information-retrieval algorithms.

The current work has adapted the earlier work by allowing for switching input streams, providing two output streams, introducing special actions, and separating the context string as a multivariate status mode from the context string component of the rule itself.

2.3 User Functions in CONIT

2.3.1 Overview

The preceding two sections described two features of the advanced CONIT system: Auto-Call and the Message Interpreter. The advanced CONIT centers on the utilization of these two features. CONIT now connects to four online retrieval systems: Lockheed DIALOG, SDC ORBIT, and two MEDLINE systems, NLM and SUNY. The user can perform all the basic functions of these several systems using the common-language features of CONIT. A considerable amount of instructional capability has also been incorporated into CONIT in order to make possible usability by a naive user.

Several improvements from the user's viewpoint have been incorporated into the present CONIT system as compared with the previous CONIT system which was described in our last report.[P6] These include:

1. A more elaborate set of instructions by CONIT at critical prespecified junctures in the program, through "canned" EXPLAIN or HELP explanations and in response to detected user errors.
2. A more complete virtual mode, especially in respect to modes of the SHOW DOCUMENTS command as well as the SHOW REVIEW and EXPLAIN DATA commands. These are discussed in detail below.
3. A more congenial user-command interface in respect to variabilities allowed in spacing, punctuation, capitalization, and abbreviation.
4. The interrupt capability.
5. Automatic connection and detaching for all systems.

These changes have been sufficient, we feel, to raise the level of the experimental interface from that of a simple demonstration of virtual system feasibility to that of an experimental vehicle for the meaningful testing of naive-user operability of a virtual-system interface.

The basic retrieval command language remains as it was described in our previous report.[P6] That report discussed the principles upon which the language was developed. A detailed description of the new CONIT system is given in the appendices. In the following subsection we outline the basic functional features of the new CONIT system as perceived by a user. This outline is structured largely by listing command syntax*.

2.3.2 System and Data-Base Selection and Connection

pick system [network]

The system named system is selected for searching, physical connections are established through the network network [or default network], and the appropriate login protocols are executed by CONIT. If another system is currently logged in, it is first logged off before system is connected. Thus,

pick dialog tymnet

connects the user to Lockheed's DIALOG system through TYMNET.

pick data database

The data base designated database is selected for searching within the currently connected system. Thus

pick data inform

connects the user to the INFORM data base.

show data

All data bases accessible to the user over the current system are listed.

---

* In this report we shall use underlining in examples of language construction to indicate variable elements and bracketing to indicate optional arguments.

explain data

For ORBIT systems, the host-system explanation of the currently selected data base is given.

explain data database

For DIALOG, the DIALOG explanation of the database designated (numbered) database is given.*

2.3.3 Additional User-Selected Instruction

explain concept

An explanation of the concept or item designated concept is given.

help

This is equivalent to "explain explain" and gives an explanation of the EXPLAIN command.

show news

The news message for the current system is given.

2.3.4 Search and Index Operations

find subject

A search is done on the subject term(s) subject. Example: find gas laser results in a search being made on the subject of gas laser.

show index subject

Those terms alphabetically surrounding the term subject are presented.

show review

All searches in the currently selected data base are shown, along with the number of documents retrieved in each.

---

* Observe this shortcoming in the degree of virtualness of the present CONIT system. It is the result of our decision at this juncture not to cope in a uniform way with the different modes by which Lockheed and SDC identify their data bases -- one uses names the other numbers. We discuss this point again, later in the report (see Section 3.3.3).

combine set i bool set k

The sets designated i and k are combined according to the Boolean operator bool, where bool = "and", "or", or "and not". Example: combine set 1 and set 2.

2.3.5 Document Reference Output

show [documents i-k] [set s] [type]

Reference information of the type type for documents numbered i through k of set s is given. The types are "title," "abstract", "citation", and "all" with the default type being "citation". The default set is the current set. The default documents are the first five. If k is missing, just document i information is output. The ordering of the arguments is optional. Examples: show documents 1-8 set 3 title yields the titles of documents 1 through 8 of set 3; while

show title

yields the titles of the first 5 document in the user's current set.

2.3.6 Interrupting and Editing

The BREAK key can be used to interrupt action of CONIT or the remote system. If CONIT action is being interrupted, control is returned immediately to the user for further commands. If remote-system action is being interrupted, then CONIT sends a simulated BREAK (series of nulls) to that system (if it accepts breaks); in any case, CONIT then waits for a signal from the remote system signifying it is ready for additional commands before returning a user cue to the CONIT user.

The MULTICS character cancel code # and line cancel code @ are used. In addition, the BREAK key can be used to kill the line and get a new user cue.

3. EXPERIMENTATION

3.1 Overview of the Experimental Program

The purpose of the experimental program described below was to give us information about the readiness of CONIT to provide a meaningful experimental environment and to give us insights to crucial issues which a full-scale experimental program should examine. Hence, the end-user sessions described below should be considered as preludes to a series of future experiments, details of which are still to be finalized.

Specifically, we sought answers to these questions: (1) In the present state of CONIT's development, can end users proceed on their own to search heterogeneous information systems and data bases without help from a specialist? (2) Is CONIT sufficiently refined to permit users themselves to extract useful information from the network of systems that CONIT interfaces? (3) On the basis of our "dry-run" experiments, what changes in CONIT should be made, if any, before more comprehensive experiments are conducted? (4) What information can be extracted from the dry-runs about formats for future experiments?

Four experimental subjects (ES's) were used in the preliminary set of experiments, and each ES was exposed to CONIT in one session. The format was along the following line. Each experimental subject was questioned about his or her previous experience with computers and retrieval systems, and the topic on which information was desired was also ascertained. This discussion consumed approximately 10 minutes. Each ES was then given a 1-to-2 minute explanation of the nature of CONIT -- not of details of use -- and our purpose in conducting the experiment.

The actual running of each session was in an analyst-monitored mode. In this mode, the analyst conducting the experiment invokes the CONIT system and presents it to the experimental subject, with the initial welcome message appearing on the typewriter-terminal printout. The ES is then asked to sit at the terminal and proceed at his own pace without further analyst help. The analyst remains at the user's side for two purposes:

(1) to record the detailed happenings of the experimental session that are not part of the user/system dialog per se (the ES is urged to "think out loud" so as to provide additional information on motivation, reactions, etc.,) and (2) to help the user in case of a system malfunction or in case the user appears so hopelessly confused or stymied that further testing would otherwise be prevented.

3.2 An Example of a User Session

In order to provide the reader with an appreciation of the detail involved in the conduct of an experimental session, we describe here the step-by-step procedure followed by one of the experimental subjects, ES3, and include interpretations of events, where appropriate. The reader may find it helpful to turn back and forth between the following discussion of ES3's session and the reproduced excerpts of it in Appendix E.

3.2.1 ES3 Background

ES3 was an M.I.T. graduate student who sought information on "Standard Methods for Chromium Labeling of Erythrocytes." ES3 had some computer experience through a statistics course that involved programming on a batch system with punched-card input. He had never operated an online terminal but he had seen it done in his previous NASIC experience which included approximately 5 problems, all involving only the MEDLINE system. Because NASIC at M.I.T. operates basically in a cooperative mode with the user present while the information specialist performs searches, ES3 had seen MEDLINE commands before. He had no experience with Lockheed or SDC procedures.

ES3 felt that his current search topic was somewhat different from his previous NASIC searches in that it was more specific -- not just on general methods, as had been the case earlier.

3.2.2 ES3 Session*

Although a reasonably good typist, ES3 exhibits some difficulty throughout the session with typing errors (see Statements U2, U3, U5, U13, U14, U19, and U22 in Appendix E) which may be attributable, in part, to unfamiliarity with the terminal keyboard. Nevertheless, he proceeds with dispatch directly down the line recommended in CONIT instruction for beginners up to the point of getting the PICK command explanation. At this point ES3 comments "I wanted an explanation of other systems and data bases"; however, he does not follow up the suggestion in this explanation (C8) to 'e systems' to get that information.

Here, the PICK DATA explanation is obtained before picking a system. At that point ES3 tears off the typescript. At this point as well as later in the session, ES3 himself expands on the notion of maintaining a a separate set of instructions for reference by tearing the typescript into several sections for individual explanations. His preference for this mode of instructional learning and review is further pinpointed when he observes "I would like to have (had) a reference manual on CONIT; I think I will make my own."

It is not clear whether the error in user request U9 (in not including the system name in the PICK command) is a typing or conceptual mistake, or some combination of the two. In any case, ES3 quickly recovers and gets connected to MEDLINE, which, he feels, is most likely the best system for his search topic.

The suggestion by CONIT in C12 to 'e explain data' confuses ES3 on several counts. First, he wonders if he had done it already (remembering perhaps, 'e pick data', U8). Second, he finds the form confusing. He finally does make the request (U12) and seems further confused and frustrated by the explanation which follows (U12). Apparently, he decides

---

* A photoreproduced copy of a section of the typescript of this session is given in Fig. 7. The excerpted dialog of the whole session -- including all user commands, is in Appendix E.

```
F CHROMIUM51

++++CONIT:
YOUR SEARCH RESULTED IN SET 4 WHICH CONTAINS 1 DOCUMENTS.
TO SEE REFERENCES TO THE FIRST FIVE DOCUMENTS TYPE: SHOW
 +++++USER::
SHOW

++++CONIT:


1
AU  - DUNN MA
AU  - GOLDWEIN MI
TITLE: HYPERSPLENISM IN ADVANCED BREAST CANCER: REPORT OF A PATIENT
       TREATED WITH SPLENECTOMY.
SO  - CANCER 35(5):1449-52, MAY 75


YOU MAY NOW SEE OTHER INFORMATION ON YOUR SEARCH RESULTS: TYPE: E SHOW
FOR EXPLANATION OF OTHER ACTIONS YOU MAY NOW DO TYPE: E COMMANDS
 +++++USER::
F CHROMIUM RADIOISOTOPES

++++CONIT:
YOUR SEARCH RESULTED IN SET 5 WHICH CONTAINS 786 DOCUMENTS.
TO SEE REFERENCES TO THE FIRST FIVE DOCUMENTS TYPE: SHOW
 +++++USER::
C SET 2 AND SET 5

++++CONIT:
YOUR SEARCH RESULTED IN SET 6 WHICH CONTAINS 203 DOCUMENTS.
TO SEE REFERENCES TO THE FIRST FIVE DOCUMENTS TYPE: SHOW
 +++++USER::
SHOW

++++CONIT:


1
AU  - FRIEDMAN H
TITLE: METHODS IN BASIC LABORATORY IMMUNOLOGY.
SO  - ADV EXP MED BIOL 73 PT B:449-55, 1976

2
AU  - UCHIDA I                                          Good
AU  - KARIYONE S
AU  - MIKI M
AU  - SATO M
AU  - FUJIMORI Y
TITLE: EVALUATION OF RECOMMENDED METHODS FOR RADIOISOTOPES RED CELL
       SURVIVAL STUDIES.
SO  - TOHOKU J EXP MED 120(3):219-23, NOV 76
```

Fig. 7 Section of the Typescript of Session for Experimental Subject ES3

that the MEDLINE data base is what he wants anyway -- or, at least, it's not worth the effort to learn about the others.

Trying to 'explain find' ES3 makes another typing error (U13). He asks the analyst how to cancel his input line and is told to use the @ key -- this is the first analyst intervention. He does this, but also types carriage return, which is not necessary and causes a USER:: prompt. Later on in the session he experiments unsuccessfully with using the key marked DEL (delete) to cancel a single character.

Methodically, collecting instructional material, ES3 gets and tears off all the explanations of how to do and refine searching (U15-U23). Notice that the spelling error of U22 ("indes" instead of "index") is automatically corrected by the CONIT command parser which only looked for the stem "ind" and ignored the (incorrect) ending.

Turning to the search itself, ES3 observes, "It doesn't appear that I have to look at a book of MEDLINE accepted terms [MESH Vocabulary]." Apparently following the search suggestions, ES3 plunges in by searching on two keywords, combining the results, and displaying output (U27, C28). This he does successfully, although he complains that CONIT's labelling of the set numbers should be more prominent so that he could more easily find them for insertion into the COMBINE command.

ES3 is somewhat dissatisfied with the four documents that are first on the set 3 output of C28. He says that the erythrocyte concept is not as central to the documents as he would like. He then follows the advice of C22 and does an index scan around the term "chromium". He wonders what the tag (TW) [text word] means but still presses on to 'show more' and uncovers two potentially valuable index terms (C31): "chromium51" (a text word) and "chromium radiosotopes" a MESH term. ES3 recognizes that these terms subsume the "labelling" concept that he wants.

After getting around a SUNY/MEDLINE time delay (C31, U31, C32), ES3 tries the first term: chromium51. This yields one document of marginal relevance (C34). He then tries the MESH term "chromium radioisotopes" (U34), combines it with the "erythrocyte" set (U35), yielding 203

documents, and shows (U36) the citation output for the first 5 documents. At least two of these (2 and 5) appear "right on the nose" -- just what he is looking for (note the annotation "Good" made by ES3, as shown in the copy of that portion of the typescript in Fig. 7).

Observing that "cell survival" appears in the title of one of the two highly relevant documents [see (2) in C37], ES3 attempts to determine if there are any more documents in his current set (6) which include that concept. He does this by searching on "cell survival" (U37), combining the result with set 6 (U38), and showing (U39) the citation information. This output (C40) shows other relevant documents but not so highly relevant as the two previously noted; in fact, not even the model document which engendered this search (document 2, set 6) is included in sets 7 or 8 since, as will be shown in C43, that document is <u>not</u> indexed under the MESH heading "cell survival".

ES3 now determines to go back to that highly relevant document and get more information. His first attempt (U40) simply repeats what he has just seen (C41 = C40). He then (U41) correctly interprets the suggestion at the end of the output response (C40, C41, etc.) and requests an explanation of the SHOW command. ES3 uses the information in C42 to request correctly what he wants (U42). After reading the abstract ES3 decides that the article is not so useful as he believed at first.

ES3 then seeks other ways (U43 - U46) to narrow the retrieved sets to only the very highly relevant documents by intersecting with another concept ("anticoagulant"). This tack produces little success and ES3 decides that the easiest approach is simply to look at all citation information for the 203 documents in set 6, which seems to include the highly relevant documents he is seeking. Rather than take the time to display this information on line, the analyst agrees to an offline dump. (A few days delay in getting the output is perfectly acceptable to ES3 since he knows that the desired information is found.) Since a command for that operation is not yet included in the CONIT virtual mode, the analyst does it in transparent mode -- i.e., using the MEDLINE language.

ES3 is satisfied at this point that he has, or will receive in offline output, all that he needs. However, the analyst encourages him to try to see if he can get more in another system. ES3 obliges by picking DIALOG (U52). He wonders whether he will automatically get disconnected from MEDLINE and notes (C53) that this is, in fact, the case.

ES3 accepts the initial data base (ERIC) in DIALOG and asks to get (U53) the explanation on the FIND command. He realizes that he has made this request before (U15) but believes that there may be different explanations for the different system and database; he realizes, on viewing C54, however, that the explanation is the same as C16. He tries the search term chromium radioisotopes (U54), gets no documents (C55), and tries to see what are appropriate terms with the SHOW INDEX command. Observe that his attempt to correct a typing error in U55 is unsuccessful but that he simply re-inputs the request correctly in U56.

Noting the index terms in C57, ES3 goes back to the original keyword strategy (U57, U59). ES3 successfully overcomes the time delay (C58) and the resulting lack of translation (C59) of search response to CONIT common-language mode.

When ES3 sees no documents (C61) from his keyword search he decides that ERIC is an inappropriate data base for him. After getting the list of DIALOG data bases (C62), ES3 attempts (U62) to get an explanation of the BIOSIS data base. Rather than continue to get the explanation he originally sought (as explained in C63), he decides to pick the BIOSIS data base but does this incorrectly (U63) by making the same name/number error that another ES did. Observing from C64 that he is still connected to ERIC, ES3 figures out his error and corrects it (U64). Observe that CONIT accepts the lack of spacing between the arguments 'data' and '5' in U64 and, nevertheless, interprets the command correctly.

ES3 then goes through the same strategy development in the BIOSIS data base as he did for ERIC (U65-U74). Recognizing that he has already seen the SHOW INDEX explanation, ES3 asks the analyst how to interrupt the output (C67). The analyst again bends the rules and explains the BREAK key. ES3 does not exercise the knowledge immediately on C67, but

he does on the very next computer message (C68). Unfortunately, the CONIT handling of the BREAK was somewhat faulty (instead of C69 it should have returned with a USER:: prompt). Nevertheless, ES3 surmounts these problems (U69-U71) and goes on with his searching.

ES3 looks at the first 5 documents (C76) in the chromium and erythrocyte set and then looks at all the documents in the subset obtained by intersecting the above set with the "labeling" concept (C80, C81). He says that these documents are all relevant but that he has already found some that are more highly relevant review-type articles in MEDLINE. He therefore decides he does not need these new ones. It is agreed to terminate the CONIT session, which ES3 does (U82).

ES3 and the NASIC information specialist, who was an observer at the CONIT session, agree that a NASIC session conducted by the information specialist is not necessary, since ES3 obtained what he needed through CONIT. ES3 feels CONIT has high potential for searchers like himself.

3.3 Interpretations of Test Usages

3.3.1 Overall Observations

The results of the preliminary testing are encouraging in that there emerged a clearcut pattern in which users demonstrated they can find the information they need through use of the CONIT interface. Furthermore, it was evident the users obtained the information they sought with little or no need for assistance other than that provided by CONIT itself. No noncomputer assistance was required by two ES's. The other two needed some help for three kinds of reasons:

1. CONIT system errors.
2. Inadequate CONIT responses to user errors (Example: failure of CONIT to ignore superfluous quotes in a command).
3. Unusual circumstances (Example: almost all communication paths to all host retrieval systems being unavailable).

Types 1 and 2 problems have already been corrected, at least for those situations that arose in the course of the experiments. The problems of Type 3 can also be alleviated, if not eliminated, as discussed later.

A compilation of data derived from the four experiments are presented in Table 1.

3.3.2 General User Psychology and Interface Design

Users are highly individualistic. Even though we have worked with only four users thus far, we have experienced the adventurons and the plodding; the careful and the careless; and the good typist and the poor typist; the easily satisfied and the not-so-easily satisfied. Whatever the category of the user, he is subject, more or less, to a pattern of user foibles such as typing errors and failure to observe, read, or follow through precisely on an instruction presented to him.

The user interface must be designed to be tolerant and understanding in the face of these variabilities and foibles. In particular, it should:

1. Provide multi-faceted approaches to allow for user variabilities.
2. Be prepared to handle a variety of expected types of user mistakes; namely, to detect when they occur and either automatically correct for them or explain to the user how to correct them.
3. Be prepared to cope with an even greater variety of mistakes it cannot fully analyze. It may do this, for example, by having generalized type error messages and instruction to help the user.
4. Avoid having a single message be so critical that missing it, or misreading it, will prevent continuation of the session; allow multiple paths and procedures to arrive at any point.

Based upon our limited use of CONIT thus far, it is apparently already doing a reasonably good job along these lines. Of course, much refinement is possible, and would be essential in order to implement a full-scale operating system.

One consistent pattern of behavior found in the experimental subjects which was not fully anticipated was their numerous requests to CONIT to print out explanatory material that they had previously requested and, presumably, read. On analysis, however, there appear to be three valid reasons for this kind of behavior:

1. Users may find it easier to have information reprinted than to search back to find it in a typescript that may be lengthy and unwieldy. Observe that users at soft-copy display terminals -- not used in

Table 1 - Summary of Experimental Uses of CONIT

Experimental Aspect	USER			
	ES1	ES2	ES3	ES4
1. Previous Computer experience:				
a. online retrieval operate	NO	NO	NO	NO
b. online retrieval observe	YES	YES	YES	YES
c. knowledge of CONIT	NONE	NONE	NONE	NONE
d. any online operating experience	YES	NO	NO	NO
e. other computer experience	MUCH	NONE	SOME	NONE
2. Search results wanted for: general interest or specific problem	GENERAL	GENERAL	SPECIFIC	SPECIFIC
3. Number of systems searched	1	1	2	1
4. Number of data bases searched	1	1	2	2
5. Session time (minutes)	23	52	104	84
6. Number of commands issued	25	26	82	70
7. Number of commands/minute	1.1	0.5	0.8	0.8
8. Number of relevant documents found	1	1	2+	3+
9. Number of document references in output	7	1	31 online 203 off-line	43
10. Number of document references copied by ES	0	0	203	8
11. Amount of analyst help required	NONE	MINOR	NONE	MINOR
12. Search satisfactory? (ES opinion)	YES	YES	YES	YES
13. CONIT satisfactory? (ES opinion)	YES	YES	YES	YES

these experiments -- having no hard-copy printout would require re-display.

2. User may not remember what they have done or seen. This is more likely to happen for new complex material where the user perceives little, if any, difference between two related requests (e.g., "explain pick data" versus "e explain data").

3. Users may sometimes suspect that the same request may yield different information, and sometimes justifiably so. This is true in CONIT now, for example, in searching different data bases. It is not true, at present, for explanatory requests; however, it would be quite appropriate, for example, to provide different explanatory information in response to the 'e find' command for different data bases. In any case, to avoid unnecessary duplicated effort and wasted time, users should be apprised when they can expect new information for the same request and how to interrupt (BREAK) repetitive output that was inadvertently requested.

Another partial answer to this problem is to provide a well-written reference manual to system use, at least for those users who fall in the category of instruction-manual readers. ES3, who was on his way to constructing such a manual on his own, was clearly in that category.

We caution, however, against putting too much faith in any one approach to instruction, no matter how reasonable it may seem a priori. Written instructional manuals, for example, appear to appeal to a narrow class of users. As previous work[16] has shown, the great majority of online users prefer to learn from the computer itself or, at least, tend to become fixated on the online medium to the exclusion of noninteractive modes. Also, there is not necessarily any greater likelihood that a user will seek what he needs, find it, and use it properly in a manual, no matter how well written, as compared with equally well designed online instruction. Then, too, in an operational online environment there is the cost and difficulty of maintaining up-to-date printed instructional material offline to all potential users, in addition to the online instruction. On the other hand, in the present cost environment, offline instruction can save on computer costs, especially if it can reduce relatively high connect-time costs.

Not only with respect to the issue of manuals versus online instruction, but more generally in the design of instruction features of the

total interface, the interface designer is faced continually with tradeoffs. One pervasive tradeoff that must be constantly faced is how much explanation to give the user at any given point in the session. There is a natural tendency, when one sees users not acting optimally because of insufficient instruction, to want to expand messages. This tendency must be carefully balanced against the deleterious effect of too much instruction. Some of our ES's complained about the bulk of the instruction. Too much explanation at any given time can result in additional complexity which, in turn, can lead the user to miss or misread or misuse instruction or, even worse, give up entirely from frustration.

That some measure of balance has been achieved in the current experimental interface is evident in that the ES's were able to learn how to use CONIT, by themselves, in the common-command languages, and they persevered until they obtained useful information from the data bases they searched. On the other hand, several features were never learned (e.g., 'show review', 'show news') or were learned only with difficulty. Upon analyzing the sessions, especially the reactions of the experimental subjects, we identified five classes of problem areas that lead to learning difficulties:

1. User Motivation. Users who fail to perceive a need for some action or who are, in general, poorly motivated, will be hindered in learning the action. Thus, a user who is satisfied with results on one data base may not want to learn about others.

2. User Foibles. We have already detailed the many ways users can err which, obviously, impede learning.

3. External Factors. Failures or deficiencies in the systems that the interface connects to can impede successful interaction and, therefore, proper learning. Bourne[17] has recently discussed related problems concerning data-base errors.

4. Interface Faults. Actual errors in, or inadequacies of, the interface and its instruction features are clearly pertinent to the learning question.

5. Instructional Tempo. The ordering and timing of instructional messages is crucial. The instructional tempo clearly relates to point 1 above, User's Motivation. As we perceive it now, the most propitious time to tell a user about a feature he has not yet learned is just at that point when he is first likely to want to use it. We are experimenting with this policy first to

determine the extent of its applicability, but extensions to what is now done in CONIT are clearly desirable: e.g., telling a user about the SHOW REVIEW command after 3 or 4 searches.

3.3.3 Data-Base Explanation and Selection

One area in which the current experimental interface fails to achieve the capabilities one would hope to realize in a good implementation of the virtual-system concept is the area of explaining to users what data bases are available, what they contain, and how to select them. In this respect the individual retrieval systems generally do only a poor-to-fair job in apprising users on line about their own data bases; at present CONIT does not fully overcome these deficiencies. A listing of possibilities in this area is given below; these improvements typify the kinds of enhancements -- from simple to very sophisticated -- that exemplify the general principles described above and that require evaluation.

1. Rearrange the response-translation rules to the SHOW DATA command so that the instructions on how to select and get explanatory information about data bases comes before the listing of the data bases. As was shown in our experiments, users get frustrated and confused when they are given a long list without first knowing what they can or should do with it. Of course, that principle must be balanced against giving a user lengthy explanations and delaying examples too long.

2. Change the instructions for EXPLAIN DATA in the SHOW DATA command response so that the user is told directly how to 'e data X' instead of having to get additional explanation first ('e e data'). This would serve the several purposes of reducing complexity and the number of interactions required, as well as avoiding potential ambiguities. (See point 2 under the Section 3.2.2.) The tradeoffs are between an increase in the length of the SHOW DATA response and additional rules, for the ORBIT case, in order to permit switching to data base X.

3. Allow the 'X' argument in the commands 'e data X' and 'pick data X', to be either the name or number of the data base. This would add to the virtualness of the interface by treating the DIALOG and ORBIT systems more uniformly as well as circumvent one predictable type of user errors. Against these advantages we must balance the cost of requiring more rules -- and, hence, a somewhat greater interface cost -- to recognize all data base names and numbers -- these numbers must be created for the ORBIT-like systems, which currently use only names.

4. A printed, offline guide to data bases (and systems) could be provided users. The tradeoffs here are the same as for other offline instruction manuals described above. We shall be experimenting with various kinds of offline guides to determine if there is any future to this approach.

5. The SHOW DATA response could be augmented to add a word or two to the tag names so as to explain the general discipline area or nature of each data base. From our experiments we would judge, tentatively, only an extra word or two is needed by users to enable them to determine whether or not the data base is relevant, or has enough relevance potential to merit getting further explanations. Tradeoffs: slight increases in response message lengths and rule size.

6. A full explanation of each data base could be maintained at the interface. If the user were urged to utilize this before connecting to the remote systems, considerable connect-hour expense could be saved. Also, limitations in data-base explanations, as well as the ORBIT data-base switching problem, could be alleviated. Tradeoff: some additional interface complexity.

7. Finally, virtualness could be extended to data bases as well as system commands by providing the interface the intelligence necessary to choose automatically the appropriate systems and data bases for a given user search topic.

3.3.4 Other Interface Considerations

Additional detailed discussions could be presented on many other areas within the scope of the interface investigations. However, we shall limit ourselves in this section to commenting on a few central issues.

The advanced experimental interface covers in virtual mode most of the basic retrieval functions. Two notable exceptions, both experienced by ES3, involve: 1) Searching on "important" terms (i.e., for documents that are importantly related to the terms they are posted under); and 2) getting offline output. Offline output is relatively easy to achieve through extension to the common command language. In an operational environment, the appropriate tagging of output by user would, of course, also be required. Specialized searching, on the other hand, may require extensions of varying degrees of complexity.

Typing mistakes, and their attendant problems appear to be pervasive. Although our experimental subjects did surmount their difficulties, it seems clear that a more definitive statement of how to connect input errors is needed in the early instruction messages. A possible alternative approach to that now being used is to suggest the use of the BREAK key followed by a redoing of the whole line after the USER prompt is received. This would simplify both editing and interrupting by making them both operative -- in a simple fashion, at least -- with the use of a single key. More sophisticated techniques could then be introduced as users became more experienced.

The Auto-Call-type connection clearly suffers from being slow, somewhat unreliable, and lacking in quick and convenient protocols for determining what connections are currently possible. It is also clear that there is much that could be done to circumvent these difficulties within the limitations of the telephone interconnections by augmenting the current message interpreter. However, it seems prudent to avoid much effort in this direction now, since emerging network improvements will soon overcome most of these problems.

4. SUMMARY

An experimental translating computer interface that provides a common, or virtual-system, mode of access to a network of heterogeneous online bibliographic retrieval systems has been developed to the point to which enhanced access by end users has been demonstrated experimentally. The interface employs a table-driven, rule-based message interpreter. The interpreter provides a dynamic, flexible means for handling the automation of the interconnection of computer and human retrieval protocols. Experimental analysis has provided preliminary evidence of success for various computer-assisted instructional techniques being employed, as well as an indication of basic tradeoffs and limitations in their implementation in working systems.

An auxiliary fallout of the project is that the interface we have explored can be used as an experimental research tool for evaluating modifications to, or enhancements of, retrieval-system features without requiring any modifications to any of the retrieval systems for which such evaluation is sought.

5. THE FUTURE OF THE VIRTUAL-SYSTEM CONCEPT

The preliminary testing of the advanced experimental CONIT interface provides a positive indication of the likely effectiveness of the translating-computer-interface/virtual-system concept. Such a concept when fully exploited should enhance the ability of an end user to access heterogeneous online retrieval systems without assistance from an intermediary. Testing of a full-scale CONIT-type interface is still needed, however, to prove the correctness of our initial conclusions and to provide the information necessary to implement an operational system.

In addition to experimentation leading to characterizations of full-scale interfaces, we believe that research is needed in order to establish the limit of potential for CONIT-like interfaces. Several research tasks can be enumerated:

1. Investigate the extent to which the interface concept can be made complete in the sense of having all features of all retrieval systems available in the virtual mode. Such a complete system should find appeal to experienced users and information specialists, as well as to inexperienced users.

2. Investigate the question: Given that the "complete" interface as defined in 1, above, is technically feasible, is it really needed to fulfill user needs? In other words, what is the cost-benefit relationship?

3. Investigate the feasibility of using interface techniques to improve the retrieval capabilities of existing systems -- for example, how can data-base selection and search-strategy formulation be automated, or computer-assisted, using techniques of the Master Index and Thesaurus type previously studied.[P2,P3]

4. Investigate the extent to which it may be desirable to extend the interface concept to functions other than bibliographic retrieval -- for example, data retrieval, fact retrieval, general data-base management, online interpersonal communication, etc.

5. Analyze the utility of the new rule-based message-interpreter with respect to any of the above extensions.

6. Examine the place of the interface concept, or parts of it, in the ultimate computer network for information transfer that may not have standardized or compatible systems.

7. Similarly, what is the role of the interface concept, if any, in the ultimate computer network for information transfer that includes standardization and compatible systems.

6. PROJECT BIBLIOGRAPHY

P1. Therrien, Charles W., Data Communications for an Experimental Information-Retrieval Network Interface, M.I.T. Electronic Systems Laboratory Technical Memorandum ESL-TM-515, August 1, 1973, NTIS Order No. PB 237 975.

P2. Marcus, R.S., "A Translating Computer Interface for a Network of Heterogeneous Interactive Information Retrieval Systems", Proceedings of the ACM Interface Meeting on Programming Languages and Information Retrieval (November 1973), SIGIR Forum, Volume IX, No. 3, Winter, 1974, Association of Computing Machinery, pp. 2-12.

P3. Reintjes, J.F. and Marcus, R.S., Research in the Coupling of Interactive Information Systems, M.I.T. Electronic Systems Laboratory Report ESL-R-556, June 30, 1974, NTIS Order No. PB 237 974.

P4. Marcus, Richard S., "Network Access for the Information Retrieval Application", Panel on Access to Computer Networks, 1975 IEEE Intercon Conference Record, Session 25/4, 1-7 (April, 1975).

P5. Marcus, R.S., "Networking Information Retrieval Systems Using Computer Interfaces", Proceeding of the 38th Annual ASIS Conference, October 26-30, 1975. (Volume 12) American Society for Information Science, pp. 77-78.

P6. Marcus, Richard S. and Reintjes, J. Francis, The Networking of Interactive Bibliographic Retrieval Systems, M.I.T. Electronic Systems Laboratory Report ESL-R-656, March 1976, NTIS Order No. PB252 407.

P7. Reintjes, J. Francis, "The Virtual-System Concept of Networking Bibliographic Information Systems", NATO Advisory Group for Aerospace Research and Development (AGARD) Conference Proceedings No. 207 on Advancements in Retrieval Technology as Related to Information Systems. Institute for Defense Analyses, Arlington, Virginia, 20-21 October, 1976.

7. REFERENCES

1. Burchinal, Lee G., "Bringing the American Revolution On-line -- Information Science and National R&D", Bulletin of the American Society for Information Science, Vol. 2, No. 8, March, 1976, pp. 27-28.

2. Overhage, C.F.J. and Reintjes, J.F., "Project Intrex: A General Review", Information Storage and Retrieval, 10 (no. 5) pp. 157-188 (1974).

3. A Brief Guid to DIALOG Searching, Lockheed Information Systems, September, 1976, Palo Alto, California.

4. ORBIT User Manual, Report TM-55 11/000/00 SDC Search Service, System Development Corporation, Santa Monica, April, 1975.

5. ERDA/RECON, User's Manual, Report TID-4586, Technical Information Center, Energy Research and Development Administration, Oak Ridge, Tennessee, December, 1976.

6. The Information Bank: An Introductory User Guid, The New York Times Information Bank, 1976.

7. On-line Services Reference Manual; Update II, Report NLM-MED-76-01, March, 1976, Bibliographic Services Division, National Library of Medicine, Bethesda, Maryland. NTIS Order No. PB 254 270.

8. Benenfield, Alan R.; Pensyl, Mary E.; Marcus, Richard S.; and Reintjes, J.F.; NASIC at M.I.T. Final Report, M.I.T. Electronic Systems Laboratory Report ESL-FR-587, February 28, 1975. NTIS Order No. PB 241 311.

9. Goldman, Jay, Auto-Call Facility, Multics Technical Note MTN-002, September 21, 1975, M.I.T. Information Processing Center Program Development Office.

10. Jordan, David M., The Multics Auto-Call Facility, MIT IPC PDO memorandum, February 27, 1976.

11. Anderson, Robert H., and Gillogly, James J., "The RAND intelligent terminal agent (RITA) as a network access aid", Proceedings 1976 National Computer Conference, AFIPS Press. pp. 501-509.

12. Davis, Randall, and Ring, Jonathan, An Overview of Production Systems, Report STAN-CS-75-524, Stanford University Computer Science Department, October, 1975.

13. Libby, R.L., Marcus, R.S., and Stallard, L.B., "An optical peripheral memory system", Proceedings of the 1967 Spring Joint Computer Conference, AFIPS Conference Proceedings Vol. 30, pp. 227-237.

14. Libby, R.L., Marcus, R.S., and Stallard, L.B., A Computing System for Non-numerical Information Processing, Technical Report 67-1373-1, Itek Corporation, February, 1967, Lexington, Massachusetts.

15. King, Gilbert W., "Table Look-up Procedures in Language Processing: Part I: The Raw Text", IBM Journal of Research and Development, Vol. 5, No. 2, April, 1961. pp. 86-92.

16. Walker, Donald E. (editor), Interactive Bibliographic Search: The User Computer Interface, AFIPS Press, Montvale, N.J., 1971. See especially, Marcus, Richard S., Benenfeld, Alan R., and Kugel, Peter, "The User Interface for the Intrex Retrieval System", pp. 159-201.

17. Bourne, Charles P., "Frequency and Impact of Spelling Errors in Bibliographic Data Bases", Information Processing and Management, Vol. 13, No. 1, 1977. pp. 1-12.

APPENDIX A

EXPLANATION OF CONTEXT STRINGS

As explained in Section 2.3, each character position in the context strings is used to identify a particular aspect of the interface context. The particular context string subcomponents that we have used to date are listed below. These explanations will be useful in understanding the applicability of the rules given in Appendix D. For each subcomponent type we list in the following table its symbolic tag (ST), the number of characters it contains (NC), the character position(s) (CP), the subcomponent name (plus explanation, if needed), followed by a listing of each subcomponent string itself.

ST	NC	CP	NAME
A	1	1	Message Sender

u: user
r: host-system response
z: user (special error condition)

B1	1-3*	2-4*	Connection Mode	(* See B2 for explanation)

not: not connected
log: logging in
con: connected
det: detaching (logging out)

B2	2	3-4	*Special Mode	(This 2 character string replaces the last 2 characters of B1 when special situation arises.)

eq: user quit (break)
eh: phone line hung up
ep: phone connection not made
ei: MULTICS I/O error detected
nr: numeral (expected while parsing message) is missing
to: timeout in waiting for host response: user not want to wait
tt: timeout in waiting for host response: control with CONIT, not user
el: timeout in waiting for host response: from DIALOG

C	1	5	Network Connected

n: none
e: Telenet
y: Tymnet

ST	NC	CP	NAME
D	2	6-7	System Connected

ll: Lockheed DIALOG
oo: SDC ORBIT
on: NLM MEDLINE
os: SUNY MEDLINE
nn: {NLM TSO (if Bl≠n)
no system (if Bl=n)

E 0-2 8-9 Retrieval Function (may be null; e.g., implicit if Bl≠c)

cb: combine
ex: explain
ed: explain data
ee: explain data fields
fd: find
pd: pick data
pk: pick
sd: show data
sh: show
si: show index
sn: show news
sr: show review

F 0-2 after E Interpretation Phase
[null]: user command parse
t: user command translate
r: host response parse and translate
ex: instructional response preparation

G 0-2 after F Sequence Number in Protocol
[alphanumeric string indicating sequencing within protocol]

A few other string subcomponents appear in special situations; e.g., in QUIT handling rules.

APPENDIX B

LISTING OF SPECIAL ACTIONS

CONIT Special Actions may be called for by the rules, as explained in Section 2.3. The various Special Actions currently employed are listed below and grouped by function.

Session Controlling

exit:	terminate session; return to MULTICS command level
exp:	call EXPLAIN routine to give explanation for concept given by next parameter
st:	call routine to set translation table to table name given by next parameter
lt:	list current translation table (starting with context given by next parameter)
rul:	call rule generating routine to add/modify/delete rule that follows as parameters to RULE command

Communication Actions

phone:	call host (network) via Auto-Call
wait:	have CONIT sleep for number of seconds given in user message
detach:	hang up Auto-Call connection

Mode Setting

setp:	set current context to character string given as next parameter
clear:	clear user and host send-message buffers
settimc:	reset continuation timeout parameter to number given in host message
settimr:	reset regular timeout parameter to number given in host message
error:	give "system error". message to user
lcft:	set flag to cause all user alphabetic input to be translated to lower case
lcff:	set flag to inhibit translation of user alphabetic input to lower case
tm4:	set mode to cause control to return to CONIT (not user) on host-response timeout
stop:	controlled exit for debug mode

Parsing Actions

orbdoc:	perform translation of SHOW DOCUMENTS parameters for ORBIT systems
skip:	skip over next character in input stream
getnum:	get numeral string appearing next in input stream and store corresponding number in variable specified in user message

putnum: put number in host message into variable specified in user message

setnumo: set number in variable specified in user message into output buffer as a character string

setnumi: set number in variable specified in user message at front of input stream

setnumu: set number in variable specified in host message into user message buffer

sub: substitute (synonym in) user message into input stream in place of current match string and retranslate.

Message Sending Actions

The following special actions cause messages to be sent to user and/or host. They are listed in a matrix in which the columns represent the particular modes in which messages are sent. If a matrix position has an "X", then the particular mode is in effect. The modes are:

SENDU: send user message
SENDH: send host message
NNL: send no new-line character at end of host message
CONT: continue translation after sending messages without getting new line for input stream
RESET: clear host input buffer before sending host message

Action	SENDU	SENDH	NNL	CONT	RESET
sendu	x				
send	x	x			
sendnnl	x	x	x		
sendc	x	x		x	
rsend	x	x			x
rsendnnl	x	x	x		x
rsnc	x	x	x	x	x

Message Building Actions

The following special actions cause messages or message components to be placed in output buffers and, possibly, messages to be sent (as indicated).

sendcr: put carriage return in host message and rsnc
tr: send input stream to host, reset first
toh: send input stream to host, do not reset
copyr: put host response in user message and sendnnl
copyrn: append next character of response line ontc user message
copyrw: append next word of response line onto user message
copyu: append rest of input onto host message and rsend
copyun: append next character of input onto host message
copyuw: append next word of input onto host message
copyf: append rest of input onto user message

APPENDIX C

EXPLAIN COMMAND MESSAGES

This appendix lists the various instructional messages a user can request currently in the experimental CONIT system. The initial "welcome" message a user sees when first entering CONIT is shown first to provide some context. Following that is listed the response to various instruction-requesting commands. These commands include the 'help' command and the 'explain' command. As explained in Section 2.3, the 'explain' command name (abbreviation: 'e') is followed by the name of some concept the user wishes the CONIT system to explain.

The listings are copied directly from a computer terminal printout so as to give the format exactly as a user would see it, except that size has been reduced by 20 percent. The start and the conclusion of a CONIT response are indicated by the lines "++++CONIT" and "*****USER::", respectively.

```
..conit
   Welcome to CONIT. For help on how to use CONIT you may type 'help'
followed by a carriage return; otherwise, you may now type any CONIT command.
 *****USER::
help

++++CONIT:
   The basic help command is 'explain' (which may be abbreviated 'e').
The command name 'explain' or 'e' must always be followed by the item
you want explained and by a carriage return.   For example, if you type
     explain concepts
or just
     e concepts
CONIT will list the basic concepts you can have explained.  The beginner
is urged to try this first.
(Remember to end every command with a carriage return.)
 *****USER::
e concepts

++++CONIT:
     EXPLANATION OF CONCEPTS
   To have a concept explained type 'explain' followed by a concept name.
Some concepts you can have explained are:
  CONCEPT NAME     CONCEPT
commands         List of CONIT commands
converse         Mechanics of the user's conversational dialog with CONIT
conit            Short background of CONIT system.
systems          Info on retrieval systems CONIT can communicate with
   The beginner is urged now to get the commands explanation by typing:
     e commands
 *****USER::
e commands

++++CONIT:
     EXPLANATION OF COMMANDS
   The following is a list of CONIT commands.
Type 'explain X' , where X is command name, for further explanation.
  NAME      ABBREV*     SHORT EXPLANATION
explain       e        Explain CONIT concepts.
pick          p        Pick a retrieval system and database to search.
find          f        Search database to find documents.
show          s        Show information on documents, data bases, etc.
combine       c        Combine sets of retrieved documents.
send                   Send command in non-CONIT language or make comment.
edit          @        Cancel your typing errors.
interrupt   BREAK      Interrupt messages or actions.
exit                   Exit from CONIT.
```

```
  *ABBREV = abbreviated form of command name
The beginner is urged to start by typing: e pick
 *****USER::
e pick

++++CONIT:
     EXPLANATION OF PICK COMMAND
   The PICK command is used to pick systems and data bases to search in.
To pick a system type 'pick X', where X is the name of the system: e.g.,
     pick medline
will get you connected to the MEDLINE system. You may also
     pick dialog
to connect to the Lockheed DIALOG system, or you may
     pick orbit
to connect to the SDC ORBIT system.
   To get more information on the retrieval systems -- and on what
to do if you can't get connected to one of them -- type
     e systems
For explanation on how to pick a data base for searching, type
     e pick data
   To logoff the current system without picking a new one, type:
     pick null
We suggest the beginnner now tear off the above instructions for reference
(if they are on paper) and then pick a system.  After you do that
CONIT will then give you further instruction.
 *****USER::
e conit

++++CONIT:
     Explanation of CONIT
   CONIT is an experimental system that connects you to different
information retrieval systems and allows you to select data bases of these
systems to search (find documents).
   CONIT will assist you in searching and you need use only the one CONIT
language to make requests of CONIT and the several systems and data bases.
The CONIT language does not yet handle all functions of every system.
Therefore, you may wish to use the language of the remote retrieval system,
if you know it, for specialized functions: for info type: e send
   CONIT has been developed under research grant from the National Science
Foundation Division of Science Information to study networking and
other improvements for interactive information systems.
 *****USER::
e find

++++CONIT:
     EXPLANATION OF FIND COMMAND
   The FIND command is used to search for documents indexed under
a particular term. Type 'find X' where X is the term you are searching for.
For example,
      find transporation
      find radiation
      find information retrieval
It is recommended that you search on single (key) words or short phrases
and then combine the resultant sets as needed.
For some additional explanations that may be useful for searching type:
     e combine
     e find more
     e show index
     e find index
     e find better
```

```
*****USER::
e show

++++CONIT:
     EXPLANATION OF SHOW COMMAND
   The  show command gives information about documents that have been found
in searching, about data bases, about index terms to search on, etc.
To have CONIT show standard citation information on some of the last set of
documents you have found type 'show'. You may also be more specific:
     show set 3 title documents 1-4
will cause the titles of the first 4 documents of set 3 to be shown to you.
     show all d 2
gives ALL information available on the SECOND document of the current set.
['d' is the abbreviation for 'document(s)'].
For more details on how to get document information type 'e show d'
Examples of other information that can be obtained are given below:
 COMMAND          BRIEF EXPLANATION
show data       Lists data bases currently available from current system
show index X    Lists terms near X in index for current data base
show review     Reviews search sets created so far on current data base
show news       Gives news from connected system
   For more details type 'e show data', etc.
 *****USER::
e combine

++++CONIT:
     EXPLANATION OF COMBINE COMMAND
   The COMBINE command allows you to make Boolean combinations of the sets
of documents you have prieviously found from searching your currently
connected data base: for example,
     combine set 2 and set 5
will make a new set which contains only documents which are in both set 2
and set 5. Similarly,
     combine set 2 or set 5
makes a new set with all documents from either set 2 or set 5. Also,
     combine set 2 and not set 5
will make a new set which contains documents in set 2 but not in set 5.
 *****USER::
e converse

++++CONIT:
     EXPLANATION OF HOW TO CONVERSE WITH CONIT
   You talk with CONIT by giving it commands. Each command consists of a
command name which may be followed by one or more additional words
to make the meaning of the command clear.
To signal the computer that you have completed your command you MUST
strike the carriage return key: the computer will just wait until you do.
   CONIT will respond to your command with some message. To signal that
its message is complete and that it is again waiting for your command
CONIT will print the 'user cue': *****USER::
You cannot give a command until you get the USER:: cue but you can
interrupt CONIT in its processing of its last command by typing the
BREAK key after which CONIT will give you a USER:: cue.
   For most errors you make CONIT will give you an informative error
message. If you detect an error BEFORE you strike the carriage return key,
you can cancel what you have typed so far by striking the BREAK key. After
the USER cue you may retype the command correctly.
(You can't use BREAK if you are connected to MULTICS via ARPANET.)
For information on other ways to edit typing errors, type: e edit
 *****USER::
```

```
e edit

++++CONIT:
     EXPLANATION OF EDITING ERRORS
   For most errors you make CONIT will give you an informative error
message. If you detect an error BEFORE you strike the carriage return key,
you can cancel what you have typed so far by striking the BREAK key. After
the USER cue you may retype the command correctly.
(You can't use BREAK if you are connected to MULTICS via ARPANET.)
   The BREAK key (also called ATTENTION (ATTN)) can also be used to interrupt
CONIT actions AFTER you request them: for info type: e interrupt
   There are two ways to cancel errors without having to BREAK and wait:
1- Type the AT SIGN (@) to cancel the command line up to that point--
then type the correct line. Thus, if you type
     shoe d@show data
you will correctly get the SHOW DATA command.
2- n NUMBER SIGNS (#) will cancel the last n characters you typed.
Thus, either of the two lines
     shoe#w data
     shoe da##w data
will also correctly give the SHOW DATA command. Combinations also work; e.g.,
     shoe d###w daat@shop dat@show dato#a
If you are connected to MULTICS thru ARPANET, you must type 2 AT SIGNS (@@)
 *****USER::
e systems

++++CONIT:
     EXPLANATION OF SYSTEMS
   There are 3 retireval systems that CONIT can connect to: 1) MEDLINE of
the National Library of Medicine; 2) Lockheed DIALOG; and 3) System
Development Corporation (SDC) ORBIT.
   MEDLINE, whose main data base is also named MEDLINE, specializes in
medicine.
Both DIALOG and ORBIT have many data bases covering all disciplines.
   If one system is not available, or to make more complete searches,
you may want to try several systems and data bases.
   There are two versions of the MEDLINE system: NLM and SUNY. The NLM
version has a few data bases not available at SUNY. For more info pick
a third system: NLM NEWS. When you 'pick medline' CONIT tries to get SUNY.
You may get NLM by typing: pick nlm
   CONIT uses either the TELENET or TYMNET network to get a system.
If a system is not available thru one network, you may try the other: e.g.,
     pick nlm telenet
 *****USER::
e interrupt

++++CONIT:
     EXPLANATION OF INTERRUPTING
   The BREAK key (also called ATTENTION (ATTN) can be used to interrupt
actions and messages before or after you request them by
simply depressing the BREAK key ONCE and waiting for CONIT to give you the
USER:: cue before making a new request. This will take a variable amount
of time depending on how long it takes CONIT to get the other system
to 'stop talking'.
   This feature may be very useful to avoid waits where, e.g., you have
already seen some instructional info, or you have seen enough document
reference output, or you are simply tired of waiting for an action to
start or be completed.
(NOTE: BREAKING occasionally causes systems to drop out or hang up.)
(You can't use BREAK if you are connected to MULTICS via ARPANET.)
```

```
 *****USER::
e exit

++++CONIT:
     EXPLANATION OF EXIT COMMAND
   The command 'exit' will log off CONIT after logging off
any system that you may currently be connected to.
   Before you exit CONIT we would appreciate any comments you may have.
For explanation type: e comment
 *****USER::
e explain data

++++CONIT:
     EXPLANATION OF EXPLAIN DATA COMMAND
   The command 'explain data' (or 'e data') gives info about data bases.
For the ORBIT and MEDLINE systems you will get information about the data
base you are currently connected to. To get information about another
data base, you must first pick it with the 'pick data' command.
   For the DIALOG system you type 'explain data X' where X is the NUMBER
of the data base as given by the 'show data' command.  For DIALOG you
do NOT have to pick the data base first.
   'explain data' for ORBIT or MEDLINE usually gives a very brief explanation
of the data base and then a detailed listing of data fields that can be used
in searching or outputing (showing document info). Unless you want to do
specialized searching or output, you may ignore the field info.
   'explain data X' for DIALOG usually gives a more lengthy synopsis of
data base X.  (For some ORBIT data bases, which are also on DIALOG,
you may get more info by typing 'e data X' on DIALOG.)
'explain data fields X' will give an explanation of the data fields
for DIALOG data base number X.
 *****USER::
e pick data

++++CONIT:
     EXPLANATION OF PICK DATA COMMAND
   To pick a data base you must first have picked and been connected
to a retrieval system.  Then you may type
      pick data X
where X is the designation of the data base you want as given by the
SHOW DATA command. For MEDLINE and ORBIT X is the data base NAME.
For DIALOG, X is the NUMBER of the data base.
 *****USER::
e find more

++++CONIT:
     FINDING MORE DOCUMENTS
   Some ways to find additional documents:
1. Use 'show index X' command to find additional search terms
2. Use 'or' in combine command to collect synonymous terms
3. Get synonymous or related terms from 'show all' document output,
   or from thesauri, dictionaries, indexes, your head, etc.
4. Break a search phrase up into single words
5. Use other data bases and systems
6. Use truncation (stemmed) searching (Type '?' after stem in DIALOG,
   ':' after stem in MEDLINE or ORBIT)
7. Drop less important terms from search: e.g, "effects" in "heat effects"
 *****USER::
e find index

++++CONIT:
```

```
     FINDING WITH TERMS FROM SHOW INDEX COMMAND
   Terms that are shown by the SHOW INDEX command may, of course,
be used as search terms in the FIND command.
   If you are connected to DIALOG, you can select particular ones
of these terms by designating their code tags instead of the full
spelled-out form of the terms.  Thus,
     find e7
will do a search on term 7 in the SHOW INDEX display, while
     find e1-e8,e12
will do a search on the first 8 terms plus term 12 (results of the
9 searches are combined with Boolean OR, as for synonyms).
 *****USER::
e find better

++++CONIT:
     FINDING BETTER DOCUMENTS
   If you have too many non-relevant documents, your search terms may
not be specific enough or you may not have included an important concept.
To correct this, search on the concept you have left out and then combine
the new search set with the previous set with a Boolean AND.
   If there is some major part of your search set that is not relevant,
and you can separate it out by some search, then you can exclude the
non-relevant part by combining the original result AND NOT the new search.
   If you know the special system commands, you can also do special searches
as by date, report number, "important terms" only, etc. E.G., in DIALOG:
     find au=marcus,r?
     find rn=nasa-cr-129524
     find amorphous(W)state/ti
Some ORBIT or MEDLINE examples:
     find hew (sa)
     find *lead and *poisoning
 *****USER::
e show document

++++CONIT:
     SHOWING DOCUMENT INFORMATION
   The basic command for showing information on documents found has the form:
     show documents 5-7 set 3 title
which causes the titles of documents 5,6, and 7 of set 3 to be output.
(Abbreviations allowed are 's' for 'show' and 'd' for 'document(s)'.)
   If you do not specify the document (d) parameter, you will get the first
5 documents, or as many as there are, if less than 5.
   If you do not specify the SET parameter, you will get the last set
you have found.
   You can get all document information by substituting 'all' for 'title'.
If you specify neither 'title' nor 'all', you will get the standard
citation information which includes title, author, and locator information.
Different systems and data bases give you somewhat different types of
information for the categories TITLE, ALL, and CITATION. This is
especially true of other-than-bibliographic data bases like FOUNDATION
GRANTS or CONGRESSIONAL RECORD.
 *****USER::
e show data

++++CONIT:
     SHOWING DATA BASES
   The 'show data' command will give you a list of all the data bases
that are available to you on the system you are currently connected to.
 *****USER::
e show index
```

```
++++CONIT:
     SHOWING INDEX TERMS
   The 'show index X' command will list terms alphabetically near to the
term X in the index for the data base you are currently connected to.
The number of documents posted under each term is also given. Thus,
     show index avogadro
will show if you have the right spelling for Avogadro and how many, if any,
documents are posted under that term, or contrarily, other related phrases
like Avogadro's number or rule or constant, etc.
   For some data bases you will also be given an indication of whether
there are listings for related thesaurus terms.
 *****USER::
e show review

++++CONIT:
     SHOWING SETS IN REVIEW
   The command 'show review' lists all the sets you have found in
searching the currently connected data base with a short description
of each set, and, for some systems, the number of documents in each set.
 *****USER::
e show news

++++CONIT:
     SHOWING NEWS
   The command 'show news' will give you the news from the system
you are currently connected to. This is in addition to the short
broadcast news you may have gotten when you first connected to the system.
   News for the MEDLINE systems is only available by picking a special
nlm news system by typing: pick nlm news
 *****USER::
e send

++++CONIT:
     EXPLANATION OF THE SEND COMMAND
   To send a command to a system in its own language you may use the
SEND command. E.g.,
     send .file 12
will connect you to DIALOG file 12 WITHOUT dropping sets just found.
     send t2/1/1-20
will give just format 1 output (doc numbers) for DIALOG set 2
     send "print indented ti, an
will give title and doc number info for MEDLINE and ORBIT.
   We welcome your comments on CONIT. To make a comment use the
'send comment' command. For explanation type: e comment
 *****USER::
e comment

++++CONIT:
     COMMENTING
   We would like to get your comments on the good and bad features
you find in using CONIT to aid us in our study of information systems.
To make a comment use the 'comment' command for each line of comment; e.g.,
     comment the common language feature is good but show is a bad name
     comment print or type is a better name than show
 *****USER::
exit

++++CONIT:
Thank you for using Conit.
```

APPENDIX D

INTERACTIVE TRANSLATION RULES

In this appendix we list the full set of approximately 500 rules for the current CONIT system. The format for the rules is explained in Section 2.2.6. The rules are grouped by function as kept in the pre-compilation file. In the regular course of updating and modifying the rules, changes are made to this file by an editor. These changes are then extracted and run on CONIT using the RUL command in a batch (absentee) mode. This absentee job causes the changes to be incorporated into the sorted file which CONIT actually interprets online. A section of the sorted file is shown in Fig. 6. Passwords have been blanked out of the rules but otherwise rules appear exactly as they are maintained in the system.

APPENDIX E

EXCERPTS FROM EXPERIMENTAL SESSION DIALOG

In Section 3.2 we discussed the CONIT computer session of experimental subject ES3. In this appendix we have transcribed excerpts from that experimental session.

During this session, there were 83 messages from CONIT to the terminal and 82 commands issued by the user at the computer terminal. The user commands are shown in full and are numbered U1 to U82. The CONIT messages, numbered C1 to C83, are sometimes shown in full; other times, for purposes of brevity, only excerpts from the CONIT messages are shown. Material left out is indicated by ellipses (...). Most of the full CONIT messages can be found in the section on EXPLAIN command messages (Appendix C) or in the section on translation rules (Appendix D). In some instances, on the other hand, output from the retrieval system itself has been excerpted to indicate just the pertinent information (see, e.g., C28, C34, etc.).

The dialog has been retyped in a format close to that which actually appeared on the original typescript. For a comparison, see Fig. 7 in Section 3.2 which is a photoreproduced excerpt of the typescript from message U32 to message C37.

```
C1:      Welcome to CONIT.  For help on how to use CONIT you may type
     'help' followed by a carriage return; otherwise, you may now type
     any CONIT command.
     *****USER::

U1:  help

C2:  ++++ CONIT:
          The basic help command is 'explain'...

U2:  e basoc cpmce@ts

C3:  ...Conit does not understand your command: "ts"
     For a list of Conit commands type: e commands

U3:  e cpmce

C4:  ...CONIT cannot yet explain cpmce
     Type 'explain concepts' for a list of concepts Conit can now explain.
     *****USER::

U4:  e concepts

C5:  ...EXPLANATION OF CONCEPTS...

U5:  e cp,,amds

C6:  ...Conit cannot yet explain cp,,amds...

U6:  e commands

C7:  ...EXPLANATION OF COMMANDS...

U7:  e pick

C8:  ...EXPLANATION OF PICK COMMAND...

U8:  e pick data

C9:  ...EXPLANATION OF PICK DATA COMMAND...
       [ES3 tears off above instructions at this point.]

U9:  p

C10: ...Conit does not understand your command: "p"...

U10: p medline

C11: +++++CONIT:
```

```
      In order to connect to the retrieval system that you have picked,
      Conit must make a telephone call.
      This will take about 30 seconds.  Please wait.
      Phone connection made to TYMNET.
      TYMNET responding.
      Logging onto suny/medline.
      News from SUNY/MEDLINE:
      6 FEB - SUNY MEDLINE AND SDILINE CONTAIN MARCH CITATIONS.
      You are now connected to SUNY/MEDLINE.
      Your initial data base is MEDLINE.
      You may now:
           (1)  See list of other data bases by typing: show data
           (2)  Find documents in this data base; for explanation type:
                e find
      *****USER::

U11:  show data

C12:  +++++CONIT:
      YOU MAY ACCESS THE MEDLINE, SDILINE, MESH VOCABULARY AND
      OLD MESH VOCABULARY FILE SETS.
      YOU ARE NOW CONNECTED TO THE MEDLINE FILE.
      To pick one of these data bases type: pick data X
          where X is the name of the data base.
      For information on what's in the data bases type: e explain data
      *****USER::

U12:  e explain data

C13:  ...EXPLANATION OF EXPLAIN DATA COMMAND...

U13:  e fomd@

C14:  *****USER::

U14:  e fomd@

C15:  *****USER::

U15:  e find

C16:  ...EXPLANATION OF FIND COMMAND...

U16:  e cp,b@

C17:  *****USER::

U17:  e combine
```

C18: ...EXPLANATION OF COMBINE COMMAND...

U18: e find more

C19: ...FINDING MORE DOCUMENTS...

U19: e sjhow index

C20: ...CONIT cannot yet explain sjhow index...

U20: e show more

C21: ...CONIT cannot yet explain showmore...

U21: e show index

C22: ...SHOWING INDEX TERMS...

U22: e find indes

C23: ...FINDING WITH TERMS FROM SHOW INDEX COMMAND...

U23: e find better

C24: ...FINDING BETTER DOCUMENTS...

U24: f chromium

C25: ...Your search resulted in set 1 which contains 499 documents. ...

U25: f erythrocytes

C26: ...set 2 which contains 10108 documents. ...

U26: c set 1 and set2

C27: set 3 which contains 30 documents. ...

U27: show

C28: ...1...TITLE: GENETIC REGULATION OF MACROPHAGE FUNCTIONS.
...
...4...TITLE: OXYGEN-DEPENDENT CIRCULATION OF SICKLE ERYTHROCYTES.
...

U28: <CR> [Carriage Return (only)]

C29: *****USER::

```
U29:  show index chromium

C30:  ++++CONIT:
      POSTINGS      TERM
           1      CHROMINUM (TW)
           1      CHROMISIZED (TW)
         386      CHROMIUM (MH)
         243      CHROMIUM (TW)
         100      CHROMIUM ISOTOPES (MH)
      To see more terms type: show more
      To select any of these terms for searching type: e find index
      *****USER::

U30:  show more

C31:  ++++CONIT
      POSTINGS      TERM
         786      CHROMIUM RADIOSOTOPES (MH)
          92      CHROMIUM-COBALT ALLOYS (MH)
           1      CHROMIUM51 (TW)
           1      CHROMIUN (TW)
           1      CHROMJAK ET (AU)
      Full response not yet received from retrieval system.
      Should CONIT wait for a response any longer?  (Answer 'yes' or 'no')
      *****USER::

U31:  no

C32:  You may now try to continue with this system or pick another
      *****USER::

U32:  f chromium51

C33:  ...set 4 which contains 1 documents. ...

U33:  show

C34:  ...TITLE: HYPERSPLENISM IN ADVANCED BREAST CANCER...

U34:  f chromium radioisotopes

C35:  ...set 5 which contains 786 documents. ...

U35:  c set 2 and set 5

C36:  ...set 6 which contains 203 documents. ...

U36:  show
```

```
C37:  ...2...TITLE: EVALUATION OF RECOMMENDED METHODS FOR RADIOISOTOPES
                    RED CELL SURVIVAL STUDIES.
      ...
      ...4...TITLE: A REAPPRAISAL OF CELLULAR LABELING WITH RADIOCHROMIUM
                    SALTS.
      ...

U37:  f cell survival

C38:  ...set 7 which contains 2562 documents. ...

U38:  c set 7 and set 6

C39:  ...set 8 which contains 42 documents. ...

U39:  show

C40:  ...1...5...

U40:  show

C41:  ...1...5...

U41:  e show

C42:  ...EXPLANATION OF SHOW COMMAND...

U42:  show set 6 all d 2

C43:  ...2
      AU  - UCHIDA T
      ...
      TITLE: EVALUATION OF RECOMMENDED METHODS FOR...
      ...
      AB  - MEAN RED CELL LIFE-SPAN IN NORMAL SUBJECTS AND IN PATIENTS WITH
            VARIOUS HEMATOLOGICAL DISORDERS WAS EXAMINED WITH 51CR AND DF32P.
            THE RESULTS WITH 51CR WERE...
      ...

U43:  f anticoagulant

C44:  ...set 9 which contains 477 documents. ...

U44:  c set 9 and set 8

C45:  ++++CONIT:
      From host: *NONE-
      *****USER::
```

```
U45:  c set 9 and set 5

C46:  ...set 10 which contains 2 documents. ...

U46:  show

C47:  ...
      TITLE: :LETTER: STUDY OF PLATELET KINETICS. ...
      [The next 5 user statements (U47-U51) were input by analyst, not ES3]

U47:  send "print offline sez##s 6 @send z#"print off-line ss 6

C48:  ...From host: NAME?

U48:  send nasic, address=mit libraries, 14s m48, citstazip=cambridge, ma
         02139

C49:  ...From host: REQUESTER'S NAME, OR SAME-

U49:  send xxxx    [actual name of ES3 input here]

C50:  ...From host: SEARCH TITLE, OR NONE-

U50:  send none

C51:  From host: OK? (Y/N/C)...

U51:  Send ok@send y

C52:  From host: OFF-LINE-PRINT COMPLETED. ...

U52:  p dialog

C53:  ...Logged off SUNY/MEDLINE.
      Logging onto DIALOG.
      News from DIALOG:
      FILE 4 IS NOW CA CONDENS VOL.86.....
      CEC (OLD FILE 4) IS NOW FILE 54
      ...

U53:  e find

C54:  ...EXPLANATION OF FIND COMMAND...

U54:  f chromium radioisotopes

C55:  ...No documents were found in your search set. ...

U55:  show n'index chromium radioisotopes
```

```
C56:  Conit did not understand the parameter 'n'index chromium
      /cradioisotopes' to the SHOW command...
      ['/c' is the MULTICS convention to indicate line continuation]
U56:  show index chromium radioisotopes

C57:  ++++CONIT:
      Ref  Index-term              Type Items RT
      E1   CHROMATOGRAPHS----------       4
      E2   CHROMATOGRAPHY----------     144  6
      E3   CHROMATORGRAPHY---------       1
      E4   CHROME------------------       2
      E5   CHROMIUM----------------      13
      E6  -CHROMIUM RADIOISOTOPES--
      E7   CHROMOBACTER------------       1
      E8   CHROMOSOMAL-------------      18
      ...
      E20  CHRONIC MEDICAL PROBLEMS          1
      To see more terms type: show more
      To select any of these terms for searching type: e find index
      *****USER::

U57:  f chromium

C58:  ...No response received from last transmission
      Should CONIT wait for a response any longer? (Answer 'yes' or 'no')
      *****USER::

U58:  yes

C59:  2From host
      From host:            2   13 CHROMIUM
      *****USER::

C60:  ...set 3 which contains 5 documents. ...

U60:  c set 3 and set 2

C61:  ...No documents were found...

U61:  show data

C62:  ++++CONIT
       1 -ERIC               2 -CA COND 70-71
       3 -CA COND 72-76      4 -CA COND 77-
       5 -BIOSIS PREVIEWS    6 -NTIS
      ...
      54 -CEC               60 -CRIS
      To pick one of these data bases type: pick data X
         where X is the number of the data base.
```

```
      For information on what's in the data bases type: e explain data
      *****USER::

U62:  e explain data biosis previews

C63:  ...EXPLANATION OF EXPLAIN DATA COMMAND...

U63:  p data biosis previews

C64:  ...Costs for use of previous data base are:
      ...
      You are now connected to:          File01:ERIC FEB77
      ...

U64:  p data5

C65:  ...You are now connected to:       File05:BIO PREVIEWS 72-77 JAN
      ...

U65:  f chromium radioisotopes

C66:  ...No documents were found...

U66:  e show index

C67:  ...SHOWING INDEX TERMS...

U67:  show index chromium radioisotopes

C68:  ...
      E3    CHROMISM----------------      28
      E4    CHROMITE----------------       6
      E5    CHROMIUM----------------    1665
      E6   -CHROMIUM RADIOISOTOPES--
      E7    CHROMO------------------     531
      ...
      E20   CHROMOC

U68:  <BREAK>  [ES hit BREAK key stopping output in middle of previous line]

C69:  Action/messages aborted; you may continue when you get the USER cue.
      From host:          -more-

U69:  f erythrocytes

C70:  Response not yet received from retrieval system.
      Should CONIT wait for a response any longer? (Answer 'yes' or 'no')
      *****USER::
      Should CONIT wait for a response any longer? (Answer 'yes' or 'no')
      *****USER::
```

U70: no

C71: ...You may now try to continue...

U71: f erythrocytes

C72: ...set 2 which contains 5222 documents. ...

U72: f chromium

C73: ...set 3 which contains 1665 documents. ...

U73: c s2 and s 3

C74: Conit does not understand your command: "combine s2 and s 3" ...
[Due to bug now fixed]

U74: c set 2 and set 3

C75: ...set 4 which contains 90 documents. ...

U75: show

C76: ...1...RADIO NUCLIDE SCANNING OF THE SPLEEN...
...5...ESTIMATION OF BONE MARROW DERIVED AND THYMUS DERIVED LYMPHO - CYTES IN LYMPHOID TISSUE BY MEANS OF PHOTOMETRY AND CHROMIUM-51 LABELED MARKER ERYTHROCYTES
...

U76: f
la@

C77: *****USER::

U77: f labeling

C78: ...set 5 which contains 3342 documents. ...

U78: c set 4 and set 5

C79: ++++CONIT:
2From host:
From host: 6 9 4*5
*****USER::

U79: s

C80: [Standard citation information for first five documents is shown]

U80: s set 6 title do @

C81: *****USER::

U81: s set 6 title documents 6 -9

C82: [Titles for documents 6-9 are shown]

U82: exit

C83: ...Logged off DIALOG. ...
Thank you for using Conit.

[Session time: 1 hour and 44 minutes = 104 minutes.]